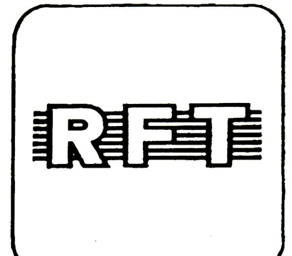


Applikation Mikrorechner

Takt

Mikroprozessor

Adressbus

Datenbus

Programm-Speicher

Daten-Speicher

Parallele Ein/Ausgabe

DMA-Steuerung

Ein/Ausgabe

Interrupt

Daten-eingabe

Daten-ausgabe

Langsame Peripherie

Daten-eingabe

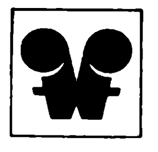
Daten-ausgabe

Schnelle Peripherie

Mikrorechnersystem mit MOS-Mikroprozessor-Schaltkreis U808D

Hardware-Beschreibung

Software-Beschreibung



KOMBINAT VEB FUNKWERK ERFURT

Anfragen und Hinweise, die den Inhalt dieser Publikation betreffen, richten Sie bitte an

KOMBINAT VEB FUNKWERK ERFURT
Applikation Bauelemente
DDR 501 Erfurt, Rudolfstraße 47, Telefon 5 87 95

Inhaltsverzeichnis Seite

Seite

Seite

Seite

Seite

Das Applikationshandbuch zum Mikrorechnersystem der 1. Leistungsklasse mit dem Schaltkreis U 808 D als ZVE gibt Anwendungshinweise zur Entwicklung von Hardware und Software solcher Systeme.
Mit der Möglichkeit des Einsatzes der Rechentechnik in Steuerungen eröffnen sich neue Prinzipien, die zu einer wesentlichen Gebrauchswertsteigerung führen und die es ermöglichen, mit einer Standardlösung der Hardware die verschiedensten konkreten Probleme im industriellen Einsatz zu lösen.
Das vorliegende Applikationshandbuch soll dem Anwender von Mikroprozessoren Kenntnisse vermitteln, die ihn in die Lage versetzen, sich in die Problematik einzuarbeiten und erfolgreich mit dem Mikroprozessorsystem der 1. Leistungsklasse zu arbeiten.
Die bedeutende Rolle der Mikroelektronik bewirkt das Erreichen von Rationalisierungseffekten in vielen Bereichen der Volkswirtschaft bei konsequenter Nutzung der mit der neuen Technik verbundenen Möglichkeiten.

1. Der Mikroprozessor U 808 D

1.1. Beschreibung des Schaltkreises

1.1.1. Allgemeines

Die integrierte Schaltung U 808 D beinhaltet eine vollständige zentrale Verarbeitungseinheit (ZVE, engl. Central Processing Unit = CPU) für den Einsatz in Mikrorechnern.
Die charakteristischen Merkmale dieses Mikroprozessors sind:

- 8-bit-Parallel-ZVE auf einem Chip in p-Kanal-MOS-Technik
- Basisbefehlssatz 48 Befehle
- vollständige innere Befehlsdekodierung und Steuerlogik
- Taktfrequenz maximal 500 kHz
- Befehlsausführungszeit 12 µs...44 µs
- TTL-Kompatibilität (Eingänge und Takt)
- Low-Power-TTL-Kompatibilität (Ausgänge)
- Verwendbarkeit von Halbleiterspeichern verschiedener Typen und Geschwindigkeiten
- direkt adressierbare Speicherkapazität 16-k-Worte (ROM oder RAM)
- Speicherkapazität unbegrenzt erweiterbar durch Speicher-Bank-Betrieb
- achtstufiger 14-bit-Adressenstapelspeicher (Stack)
- sieben frei verfügbare Datenregister
- Unterbrechungsmöglichkeit (Interrupt)
- 18poliges DIL-Plastgehäuse.

Der U 808 D ist ein 8-bit-Mikroprozessor auf einem Chip in p-Kanal-Silicon-Gate-Technik. Der Prozessor kommuniziert über einen 8-bit-Daten- und Adreßbus (D0 bis D7) und besitzt die zwei Steuereingänge READY und INTERRUPT (Bereit und Unterbrechen) und vier Ausgänge (S0, S1, S2 und SY). Steuerinformationen, 14-bit-Adressen und Daten werden im Zeitmultiplex zwischen der ZVE und den externen Speichern übertragen. Die ZVE enthält sechs 8-bit-Datenregister, einen 8-bit-Akkumulator, zwei 8-bit-temporäre-Register, vier Bedingungs-Flipflops und eine binäre 8-bit-Parallel-Arithmetikeinheit zur Ausführung von arithmetischen und logischen Operationen. Ein interner Stapelspeicher (Stack) beinhaltet einen 14-bit-Programmzähler und sieben 14-bit-Adressenspeicher zum

Abspeichern von Programm- und Unterprogrammadressen.
Die 14-bit-Adressen ermöglichen die direkte Adressierung von 16 k 8-bit-Worten (Bytes) im externen Speicher.
Der Steuerteil des Chips enthält die Logik zur Steuerung der funktionellen und zeitlichen Abläufe der internen Baugruppen.
Der Befehlssatz der ZVE umfaßt 48 Basisbefehle.
Die Befehle unterteilen sich in 1-Wort-, 2-Wort- und 3-Wort-Befehle. 1-Wort-Befehle werden für arithmetische und logische Operationen und zum Datentransfer verwendet. 2-Wort-Befehle verknüpfen immer unmittelbar mit dem Befehl ein Datenwort und 3-Wort-Befehle dienen der Ausführung von Sprung- und Rufoperationen.
Der normale Programmablauf der ZVE kann durch ein Unterbrechungssignal am INTERRUPT-Eingang (IT) unterbrochen werden.
Dadurch wird z. B. die Bedienung von peripheren Geräten während der Bearbeitung des Hauptprogramms möglich. Der READY-Eingang (RD) ermöglicht das Arbeiten der ZVE mit Speichern verschiedener Typen und Zugriffszeiten. Die Status-Ausgänge (S0, S1, S2) und der Synchronisationsausgang (SY) zeigen zu jedem Zeitpunkt den Arbeitszustand der ZVE innerhalb eines Befehlszyklus an. Durch diese Ausgänge wird die Steuerung der gesamten äußeren Baugruppen des Mikrorechners gewährleistet. Der Datentransfer innerhalb der ZVE und zwischen ZVE und äußeren Baugruppen erfolgt bidirektional auf einem Datenbus mit der Breite von 8 bit.
Über diesen Datenbus sendet oder empfängt die ZVE Befehle, Adressen, Daten und Steuersignale in Form von 8-bit-Wörtern.
Alle Eingänge der ZVE sind TTL-kompatibel (TTL-Ausgang mit Widerstand an TTL-Betriebsspannung) und mit integrierten Schutzschaltungen versehen. Die Ausgänge der ZVE sind Low-Power-TTL-kompatibel (TTL-Eingang mit Eingangs-Clampingdiode).
Für den U 808 D sind logisch 1 als H- und logisch 0 als L-Pegel definiert.

1.1.2. Zustandskodierung

Der Prozessor steuert die Arbeitsweise des Datenbusses, ob Daten gesendet oder empfangen werden. Die Zustandssignale S0, S1, S2 informieren im Zusammenwirken mit dem SY-Signal die externe

Steuerlogik über den aktuellen Zustand der ZVE.
Insgesamt werden acht mögliche Zustände der Befehlsabarbeitung unterschieden (Tafel 1.1).

Tafel 1.1: Zustandskodierung des U 808 D

S0	S1	S2	Zustand
0	1	0	T1
0	1	1	T1I
0	0	1	T2
0	0	0	T3W
1	0	0	T3
1	1	0	T3S
1	1	1	T4
1	0	1	T5

1.1.3. Zeitlicher Ablauf der Befehlsabarbeitung

Ein interner Maschinenzyklus besteht aus maximal fünf Zuständen (T1 bis T5), davon zwei Zustände, in welchen Adressen oder Daten nach außen gesendet werden (T1 und T2), einen Zustand für das Holen von Befehlen oder Daten bzw. Senden von Daten (T3) und zwei Zustände für die Befehlsausführung (T4 und T5). Jeder Zustand umfaßt zwei Takte C1 und zwei Takte C2.
Für das zeitlich richtige Zusammenwirken des Prozessors mit dem externen Speicher kann bei Verwendung von Halbleiterspeichern mit langen Zugriffszeiten der READY-Eingang in Verbindung mit dem Wartezustand WAIT (T3W) zur Synchronisation genutzt werden.
Der READY-Eingang wird am Ende des Zustandes T2 in jedem Maschinenzyklus von der Prozessorsteuerung abgefragt. Liegt der Eingang zu diesem Zeitpunkt auf log. 0 (Speicher noch nicht bereit), geht der Prozessor in den Wartezustand WAIT.
Dieser wird verlassen, wenn der READY-Eingang auf log. 1 (Speicher bereit) geschaltet wird.
Ein externes Interrupt-Signal wird vom Prozessor durch den Zustand T1I quittiert. Dieser Zustand entspricht in seiner Wirkung nach außen dem Zustand T1, nur lag vorher eine Unterbrechungsanmeldung (IT = 1) vor.

Der Zustand STOP (T3S) wird eingenommen, wenn der Prozessor einen Halt-Befehl dekodiert.
Bild 1.1 zeigt den Zeitablaufplan für einen typischen Prozessor-Maschinenzyklus.

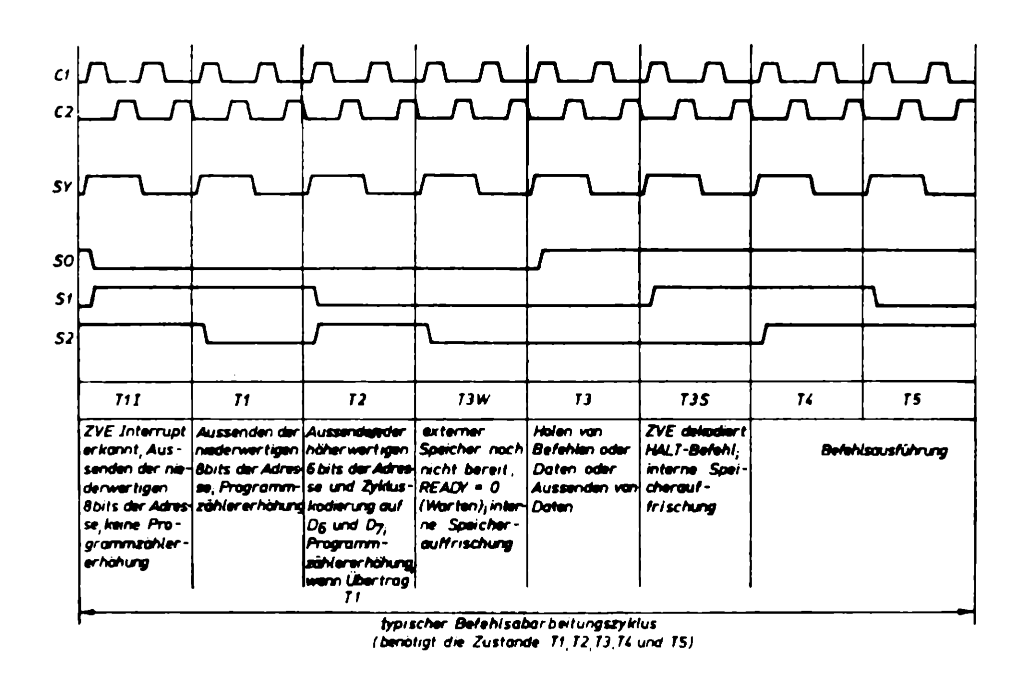


Bild 1.1: Zeitablaufplan für einen Befehlsabarbeitungszyklus

Bild 1.2 zeigt das Zustands-Flußdiagramm für die möglichen Maschinenzyklen.

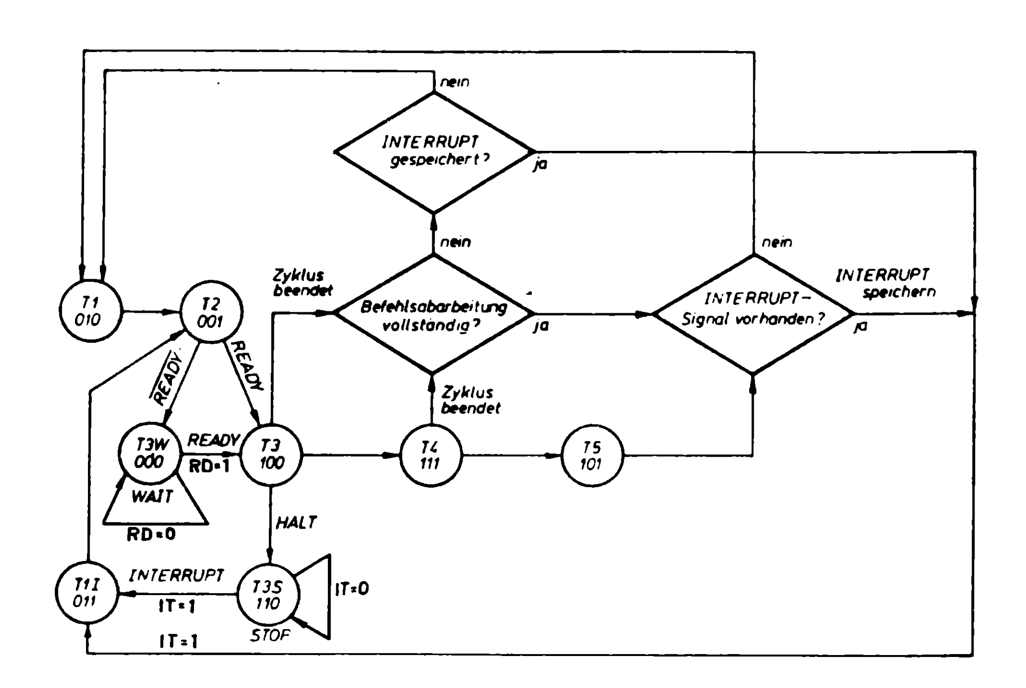


Bild 1.2: Zustands-Flußdiagramm des U 808 D

Für die Befehlsabarbeitung **benötigt die ZVE minimal einen und** maximal drei Maschinenzyklen. Einige Befehle benötigen in bestimmten Zyklen die Zustände T4 und T5 nicht. In diesen Fällen werden diese Zustände übersprungen. Die Zustände WAIT und STOP können beliebig lang sein und werden von der ZVE erst verlassen, wenn die entsprechenden Signale dafür anliegen (RD = 1, IT = 1). Für die Realisierung der Unterbrechungsmöglichkeit enthält die ZVE ein Unterbrechungs-Flip-Flop zur internen Zwischenspeicherung des Unterbrechungssignales.
Nach Beendigung eines jeden Befehles und nach Beendigung eines jeden Zyklus eines Befehles **wird das Unterbrechungs-Flip-Flop** von der ZVE-Steuerung abgefragt und in Abhängigkeit von seiner Lage entschieden, ob der Zustand T1 oder der Zustand T1I eingenommen wird.
Erreicht ein Interrupt-Signal die ZVE, so übernimmt das Unterbrechungs-Flip-Flop das Signal mit Beginn des Zyklus 1 des nachfolgenden Befehls. Die ZVE arbeitet somit den gerade in Durch-

führung befindlichen Befehl erst vollständig ab, ehe auf das Interrupt-Signal reagiert wird. Mit Beendigung des letzten Zyklus des Befehls, der wegen des Interrupt-Signals die ZVE erreicht, wird das interne Unterbrechungs-Flip-Flop zurückgesetzt und für eine erneute Unterbrechung vorbereitet.

1.1.4. Zykluskodierung

Die Zykluskodierung dient der externen Kennzeichnung des gerade in Abarbeitung befindlichen Zyklus in der ZVE. Die ZVE kann maximal vier verschiedene Zyklen unterscheiden.
Tafel 1.2 zeigt die Zykluskodierung und deren Bedeutung.

Tafel 1.2: Zykluskodierung des U 808 D

D_6	D_7	Zyklus	Funktion
0	0	PCI	ausgesendete Adresse für Lesen aus dem Speicher (1. Byte des Befehls)
0	1	PCR	ausgesendete Adresse für Lesen aus dem Speicher (2. und 3. Byte des Befehls)
1	0	PCC	ausgesendete Daten und Adressen für I/O-Operation
1	1	PCW	ausgesendete Adresse für Dateneinschreiben in den Speicher

Der erste Zyklus eines jeden Befehls ist immer ein Befehlholzyklus (PCI). Der zweite und der dritte Zyklus sind immer ein Datenlesezyklus (PCR), ein Datenschreibzyklus (PCW) oder eine Eingabe-Ausgabe-Operation (Input/Output = I/O-Operation) (PCC).
Die Zykluskodierung ist mit zwei Bits realisiert und wird in jedem Zyklus während T2 auf dem Datenbus an den Stellen D6 und D7 ausgegeben.

1.2. Innerer Aufbau der ZVE

Die wichtigsten Funktionsgruppen der ZVE sind das Befehlsregister mit dem Befehlsdekoder, die Speicher, die Arithmetik-Logik-Einheit (engl. Arithmetic Logic Unit = ALU) und der Datenbus-Puffer. Der Datentransport zwischen den einzelnen Baugruppen innerhalb der ZVE erfolgt über den internen Datenbus. Das Blockschaltbild des inneren Aufbaus der ZVE zeigt Bild 1.3.

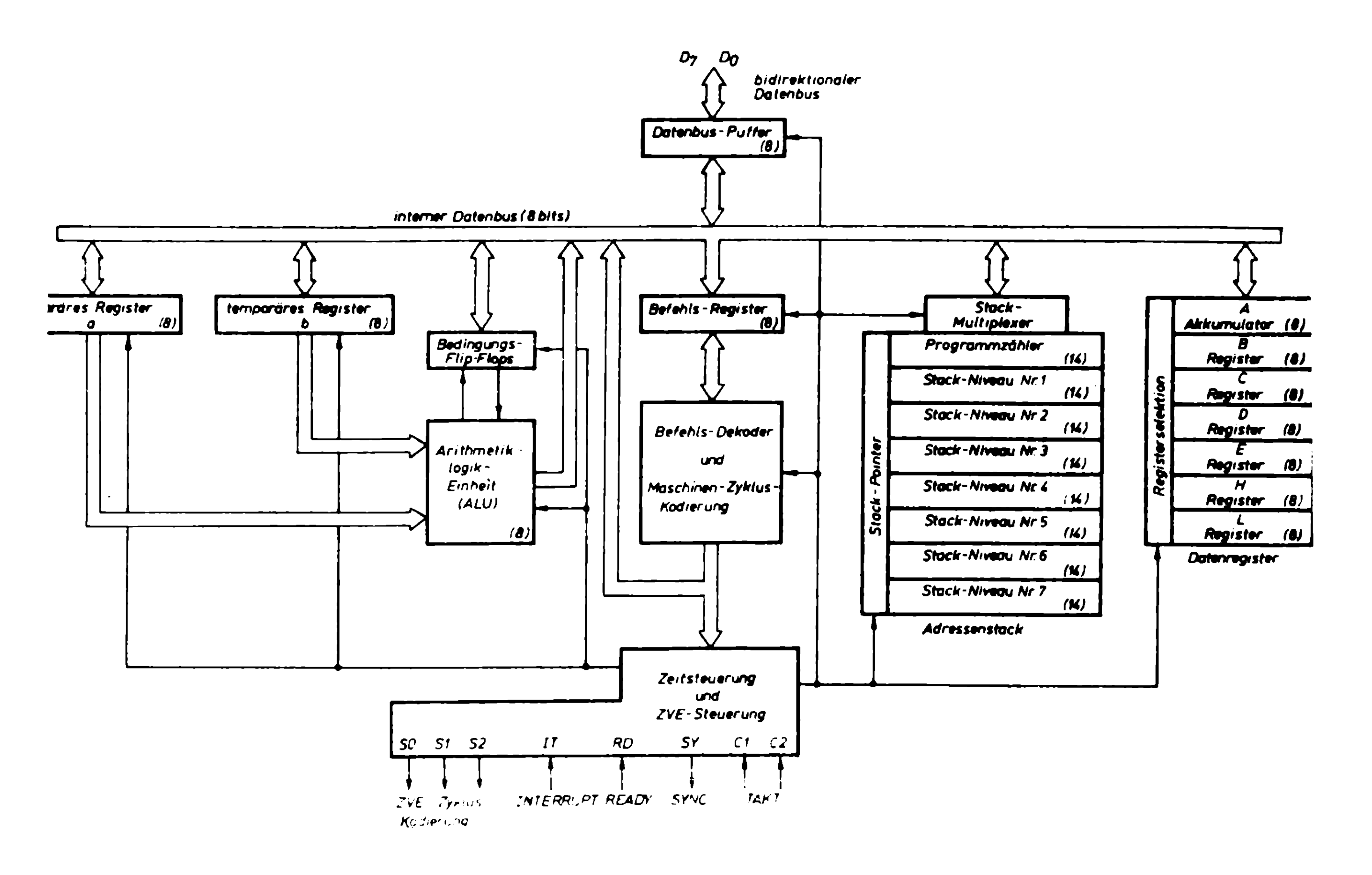


Bild 1.3: Innerer Aufbau des U 808 D

1.2.1. Befehlsregister und Dekoder

Das Befehlsregister und der Befehlsdekoder bilden den Kern der Prozessorsteuerung. Die vom externen Programmspeicher gelieferten Befehle werden im Befehlsregister gespeichert und für die Steuerung der internen Speicher und der ALU im Befehlsdekoder dekodiert.

Werden bei der Befehlsabarbeitung nicht alle Zustände benötigt, so steuert der Befehlsdekoder auch den verkürzten Zyklusablauf.

1.2.2. Speicher

In der ZVE werden ein achtstufiges 14-bit-Adressenstack und ein siebenstufiger 8-bit-Datenspeicher (Notizspeicher) verwendet.

Adressenstack

Der Adressenstack (Stapelspeicher) beinhaltet acht 14-bit-Register zur Speicherung von 8 bit des niederwertigen Adreßteils und 6 bit des höherwertigen Adreßteils (zur Adressierung von 16 k Bytes = 2^{14} Bytes werden 14 Adreßbits benötigt). Ein Register des Stacks wird als Programmzähler verwendet. Im Programmzähler steht immer die Adresse des nächsten zu verarbeitenden Befehls. Die anderen sieben Register des Adressenstacks dienen der Abspeicherung von maximal sieben Rückkehradressen bei Aufruf von Unterprogrammen und Interrupt-Service-Routinen.
Der Stack arbeitet als LIFO-Stack (Last In/First Out). Er speichert automatisch den Inhalt des Programmzählers bei der Ausführung eines Rufbefehls und gibt den zuletzt gespeicherten Wert bei Ausführung eines Rückkehrbefehls an den Programmzähler ab.
Der Stack wird von einem 3-bit-Stack-Pointer verwaltet, der seinen Füllstand anzeigt. Wird die Kapazität des Stacks überschritten, so läuft der Stack-Pointer über, und die zuerst gespeicherte Rückkehradresse (Hauptprogramm) wird vernichtet.
Die niederwertigen 8 bit des Programmzählers werden unmittelbar nach dem Aussenden des niederwertigen Adreßteils in T1 erhöht, die höherwertigen 6 bit unmittelbar nach Aussenden des höherwertigen Adreßteils in T2, wenn von T1 ein Übertrag resultiert.
Die vom Programmzähler maximal adressierbaren 16 k byte Speicherkapazität können durch Anwendung des Speicherbankbetriebes (hardwaremäßige Selektion aus mehreren 16 k byte Speicherblöcken durch zusätzlichen Ausgabebefehl) praktisch unbegrenzt erweitert werden.

Datenspeicher (Indexregister)

Der Datenregisterblock beinhaltet einen 8-bit-Akkumulator (A) und sechs weitere 8-bit-Register (B, C, D, E, H, L). Alle arithmetischen Operationen verwenden den Inhalt des Akkumulators als einen Operanden.

Die Register sind unabhängig voneinander und können für Zwischenspeicherungszwecke verwendet werden. Die Register H und L dienen gleichzeitig als Adressengenerator für eine externe Speicheroperation (indirekte Adressierung). Für den Fall einer externen Speicheroperation (Lesen oder Schreiben) enthalten Register H den höherwertigen 6-bit-Adreßteil und Register L den niederwertigen 8-bit-Adreßteil des gewünschten externen Speicherplatzes. Die Bits D_6 und D_7 von Register H werden dabei nicht benötigt und sind wahlfrei.

1.2.3. Arithmetik-Logik-Einheit (ALU)

Die arithmetischen und logischen Operationen (Addition und Subtraktion mit und ohne Übertrag, logisches UND und ODER, exklusives ODER, Vergleich, Erhöhung und Erniedrigung) werden in der 8-bit-Parallel-ALU durchgeführt. Zwei temporäre Register a und b speichern die beiden zu verarbeitenden Operanden kurzzeitig in den internen Datentransferphasen ab.
Gleichzeitig dienen diese Register auch als Zwischenspeicher beim Datentransfer von Adressen und Daten mit externen Baugruppen des Mikrorechners.
Von der ALU werden bei bestimmten Operationen in Abhängigkeit vom Ergebnis vier Bedingungs-Flip-Flops (Flags) gesetzt.
Die Bedingungs-Flip-Flops liefern die Möglichkeit der bedingten programmtechnischen Verzweigung bei Ruf-, Sprung- und Rückkehroperationen. Sie haben folgende Bezeichnungen:

C (Übertrag) = Ergebnis der Operation ergibt einen Übertrag
Z (Null) = Ergebnis der Operation ergibt Null-Inhalt
S (MSB) = Ergebnis der Operation ergibt MSB = 1 (höchstwertiges Bit = 1)
P (Parität) = Ergebnis der Operation ergibt Parität des binären Inhaltes.

1.2.4. Datenbus-Puffer

Der Datenbus-Puffer stellt die Verbindung des internen mit dem externen Datenbus her. Der Puffer besteht aus acht bidirektionalen Ein- und Ausgangsstufen, deren Zustand von der ZVE gesteuert wird. In Bezug auf die Anschlußkompatibilität ist der Datenbus-Puffer mit den anderen Eingängen und Ausgängen der ZVE identisch.

1.3. Befehlssatz des U 808 D

Die Abspeicherung der Befehle und Daten im externen Programmspeicher erfolgt in 8-bit-Wörtern (Bytes) der Form

$$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1\ D_0$$

D_7 entspricht dem höchstwertigsten (2^7) Bit und D_0 dem niederwertigsten (2^0) Bit. Die Befehle haben die Länge von ein, zwei oder drei Worten. Die Bytes von Mehrwortbefehlen müssen im externen Speicher aufeinanderfolgende Adressen haben.
Tafel 1.3 zeigt die drei Befehlsformen und ihre Verwendung.

Tafel 1.3: Befehlsformen des U 808

Befehlsform		typische Befehle
Ein-Wort-Befehle		Register-Lade-Befehle, Memory-Operationen, I/O-Befehle, arithmetische u. logische Befehle, Verschiebebefehle und Rückkehrbefehle
$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1\ D_0$	Befehlskode	
Zwei-Wort-Befehle		Befehle, die unmittelbar Daten verknüpfen
$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1\ D_0$	Befehlskode	
$B_2\ B_2\ B_2\ B_2\ B_2\ B_2\ B_2\ B_2$	Operand	
Drei-Wort-Befehle		Sprung- oder Rufbefehle
$D_7\ D_6\ D_5\ D_4\ D_3\ D_2\ D_1\ D_0$	Befehlskode	
$B_2\ B_2\ B_2\ B_2\ B_2\ B_2\ B_2\ B_2$	niederwertiger Adreßteil	
$X\ X\ B_3\ B_3\ B_3\ B_3\ B_3\ B_3$	höherwertiger Adreßteil	

In der Tafel A 1.1. (Anhang) sind die 48 Grundbefehle der ZVE beschrieben.
Gleichzeitig werden die Mnemonic-Kode, die minimal notwendigen Grundzustände und die binären Befehlskode mit angegeben. Die 48 Basisbefehle lassen sich durch Anwendung auf die verschiedenen Register, Tore und Bedingungs-Flip-Flops zu einem Befehlssatz von 226 in ihrer Wirkung verschiedenen Befehlen erweitern.

Detaillierte Befehlserläuterung

Die Tafeln A 1.2.1. bis A 1.2.10. beschreiben die einzelnen Befehle und ihre genaue Wirkung.
Es wird folgende Symbolik verwendet:

A = Akkumulator (Indexregister A)
A_n = Bit n des Akkumulatorinhaltes" $0 \leq n \leq 7$
ADR = beliebige Memoryadresse
C = Bedingungs-Flip-Flop: Übertrag
DATEN = beliebiges Datenbyte
DST = Bestimmungsregister oder Bestimmungsmemorybyte
EXP = Konstante, mathematischer Ausdruck
M = ein Memorybyte (ein ZVE-externer Byte-Speicherplatz)
P = Bedingungs-Flip-Flop: Parität
PC = Programmzähler
REGM = beliebiges Register oder Memorybyte
S = Bedingungs-Flip-Flop: MSB
SRC = Quellenregister oder Quellenmemorybyte
STK = oberstes Stack-Register
Z = Bedingungs-Flip-Flop: Null
( ) = Inhalt des Registers oder des Memorybyte
$\longleftarrow$ = wird ersetzt durch

1.4. Prozessor-Steuersignale

Der U 808 D benötigt als externe Steuersignale die Taktsignale C1 und C2, das INTERRUPT- und das READY-Signal.

1.4.1. Takt

Der Takt besteht aus zwei nichtüberlappenden Taktsignalen C1 und C2. Für jeden Zustand benötigt die ZVE zwei Taktimpulse.
Der Takt C1 wird generell zum Laden aller Datenwege und inneren Speicher und C2 für den Datentransfer in den Prozessor verwendet.
Das SY-Signal wird durch 1:2-Teilung von C2 abgeleitet und von der ZVE zur externen Steuerung ausgesendet.

1.4.2. INTERRUPT (IT)

Das INTERRUPT-Signal ermöglicht einerseits eine Programmunterbrechung eines laufenden Programms und andererseits das Verlassen des STOP-Zustandes, der durch das Einlesen eines HALT-Befehls herbeigeführt wurde.
Liegt das INTERRUPT-Signal an (IT = 1), so wird beim nächsten Datenholezyklus (PCI) über die Statusausgänge im Zustand T1I die Kodierung S0S1S2 = 011 ausgesendet. In T1I und dem anschließenden T2 werden der momentane Programmzählerzustand ausgesendet, nach dem Aussenden jedoch nicht erhöht. Während des nachfolgenden Zustandes T3 kann ein beliebiger Unterbrechungsbefehl über einen an den ZVE-Datenbus angekoppelten Eingangsmultiplexer in die ZVE eingelesen und damit auf das INTERRUPT-Signal reagiert werden (z. B. HLT, RST, JMP oder CALL). Enthält der Befehlskode des eingelesenen Befehls die Anweisung, daß es sich um einen Mehrwortbefehl handelt (z. B. JMP oder CALL), so nimmt die ZVE selbsttätig während der gesamten zum Befehl gehörenden Zyklen für den Zustand T1 den Zustand T1I ein, ohne daß noch einmal ein INTERRUPT-Signal gegeben werden muß.
In jedem Zustand T1I wird die entsprechende Zustandskodierung (S0S1S2 = 011) ausgesendet, so daß eine externe Logik für alle Bytes des INTERRUPT-Mehrwortbefehls den Eingangsmultiplexer geöffnet halten kann. Ist der Befehl vollständig abgearbeitet, setzt die ZVE mit dem Programmzählerstand, der ihr durch den INTERRUPT-Befehl vorgegeben wurde (z. B. Startadresse einer Interrupt-Service-Routine), die Programmabarbeitung fort.

Tafel 1.4 zeigt das Prinzip für eine solche Interrupt-Service-Routine-Behandlung.

Tafel 1.4: Prinzip der Unterbrechungsbehandlung

Adresse	Befehl	PC-Stand	Bemerkung
N - 1	Befehl N - 1	N	INTERRUPT-Signal erreicht Eingang IT der ZVE
N	Befehl N	N + 1	Aufruf der Interrupt-Service-Routine mit Restart oder CALL. PC-Stand (Größe N) wird in den Adressenstack übernommen (Erhöhung um ein Stack-Niveau).
N + 1	Befehl N + 1	N + 2	
.	. .		
.	. .		
.	. .		
S	Befehl S	S + 1	
S + 1	Befehl S + 1	S + 2	
S + 2	Befehl S + 2	S + 3	Nach Ausführung der Unterroutine kehrt die ZVE zu dem Programm zurück, das durch das INTERRUPT-Signal unterbrochen wurde. Der PC-Stand wird vom Adressenstack bestimmt (Erniedrigung um ein Stack-Niveau).
.	. .	.	
.	. .	.	
.	. .	.	
.	. .	.	
.	. .	.	
.	. .	.	
.	. .	.	
.	. .	.	
		N	
S + K	RETURN		

Durch das INTERRUPT-Signal wird auch die Programmabarbeitung nach Einschalten der Speisespannung der ZVE gestartet.
Mit dem Einschalten der Speisespannung wird die ZVE automatisch in den STOP-Zustand geschaltet. In den folgenden 16 Taktperioden nach Einnehmen dieses Zustandes werden alle ZVE-internen Register und Speicher gelöscht. In dieser Zeit darf kein INTERRUPT-Signal an der ZVE anliegen (IT = 0). Wird nach Beendigung der 16 Taktperioden INTERRUPT gegeben, so verläßt die ZVE den STOP-Zustand, und die Programmabarbeitung wird gestartet.
Für den Programmbeginn gibt es mehrere Varianten, die in Tafel 1.5 aufgeführt sind.

Tafel 1.5: **Startvarianten** des U 808 D

A: Start durch den normalen Programmablauf

Durch ZVE ausgesendete Adresse	Befehl im ROM	Bemerkung
000000 00000000	**MOV A, A**	Direkter Eintritt in
000000 00000000	**MOV A, A**	das Hauptprogramm. Der
000000 00000001	**Befehl 1**	erste Befehl wird wegen
000000 00000010	**Befehl 2**	der nicht erfolgten
. .	.	PC-Erhöhung im Zustand
. .	.	T1I zweimal geholt.
. .	.	

B: Start durch **programmierten Sprung** ins Hauptprogramm

Durch ZVE ausgesendete Adresse	Befehl im ROM	Bemerkung
000000 00000000	RST	(RST = 00AAA101)
000000 00AAA000	Befehl 1	Der erste Befehl (RST)
000000 00AAA001	Befehl 2	bewirkt einen Sprung
. .	.	in das Hauptprogramm.
. .	.	Der bei RST in den Stack
. .	.	gerettete PC-Stand wird
		nicht weiter verwendet.

C: Start durch erzwungenen Restart

Durch ZVE ausgesendete Adresse	Befehl im ROM	Bemerkung
000000 00000000	Befehl 1	Statt Befehl 1 erfolgt
		erzwungener Restart
		(RST)
000000 00AAA000	Befehl a	Sprung in eine Start-
000000 00AAA001	Befehl b	Routine. PC-Stand wird
. .	.	in den Stack gerettet.
. .	.	
. .	.	
000000 00nnnnnn	RET	Rückkehr aus der Start-
000000 0000000	Befehl 1	routine und Ausführung
. .	.	des ersten Befehls.
. .	.	
. .	.	

1.4.3. READY (RD)

Der Prozessor U 808 D wird durch der Steuereingang READY befähigt, mit beliebigen Halbleiterspeichertypen und Geschwindigkeiten zusammenzuarbeiten. Liegt der READY-Eingang im Zustand T2 während der Taktzeit C22 auf 0, so nimmt die ZVE unmittelbar nach T2 den Zustand WAIT ein und verbleibt dort so lange, bis der READY-Eingang auf 1 gelegt wird.
Im Zustand WAIT bleiben alle momentanen Speicherinhalte und Flip-Flop-Zustände der ZVE erhalten, es erfolgt ein kontinuierliches Auffrischen (Refresh) der dynamischen Speicher.
Bei Verwendung von schnellen Speicherelementen kann der READY-Eingang fest auf 1 gelegt werden, wodurch ein Wartezustand zwischen T2 und T3 unterdrückt wird. Der READY-Eingang kann auch für die Realisierung von Einzelschrittbetrieb der ZVE verwendet werden. Hierfür ist es notwendig, den READY-Eingang in jedem Zyklus auf 0 zu legen, so daß die ZVE grundsätzlich den Zustand WAIT einnimmt. Für jeden Einzelschritt (ein Zyklus) ist nun der READY-Eingang kurz umzuschalten.

1.5. Technische Kennwerte

1.5.1. Eingänge

Die Anschlüsse C1, C2, RD und IT sind reine Eingänge. Sie sind mit einem gesteuerten Widerstand zwischen Eingang und U_{CC} (+5 V) versehen, um den TTL-H-Pegel zu verbessern (Bild 1.4). Diese Eingänge können nicht mit MOS-Schaltkreisen angesteuert werden.

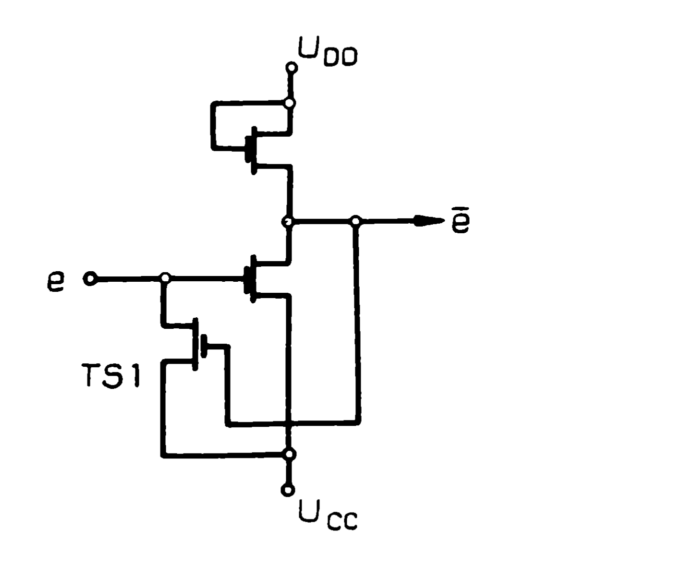


Bild 1.4: Eingangsschaltung des U 808 D

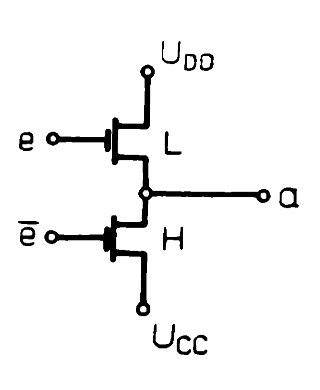


Bild 1.5: Ausgangsschaltung des U 808 D

1.5.2. Ausgänge

Reine Ausgänge sind die Anschlüsse SO, S1, S2 und SY. Die Ausgänge sind wie Gegentaktendstufen konzipiert (Bild 1.5).
Da der L-Transistor gegen U_{DD} (-9 V) arbeitet, können an diesen Ausgängen negative Spannungen auftreten, die von TTL-Eingangsdioden begrenzt werden müssen.
Der L-Transistor ist so dimensioniert, daß er Low-power-TTL-Schaltkreise treiben kann (Lastfaktor = 1/4 TTL-Last).

1.5.3. Ein- und Ausgangsstufen

Die Bustreiber D0 bis D7 stellen kombinierte Ein- und Ausgangsstufen dar (Bild 1.6).

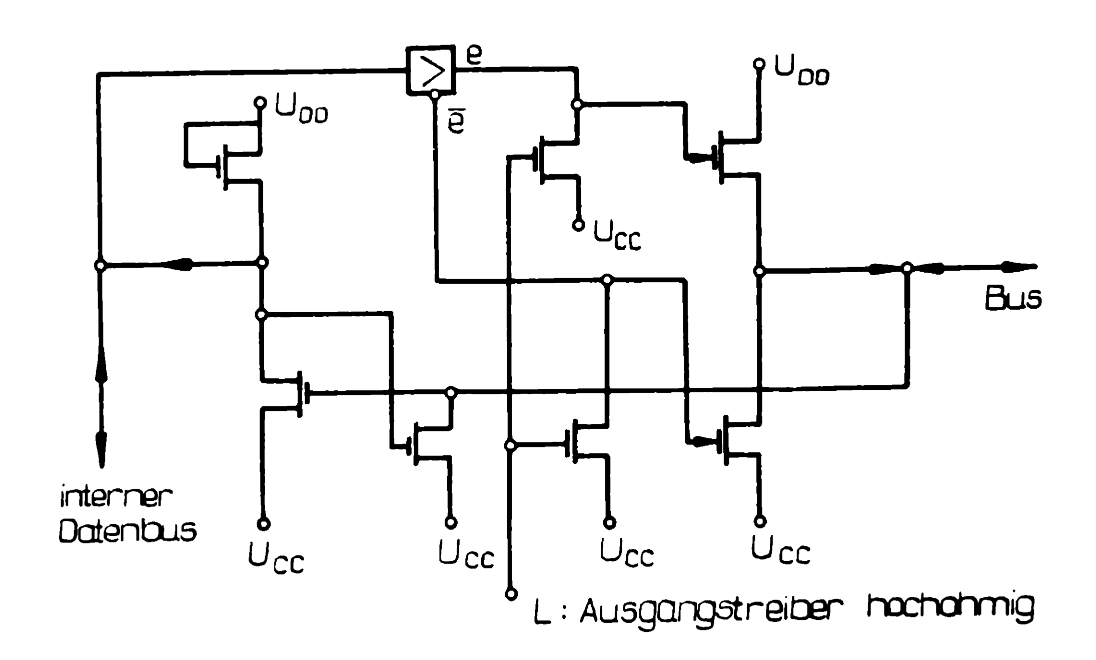
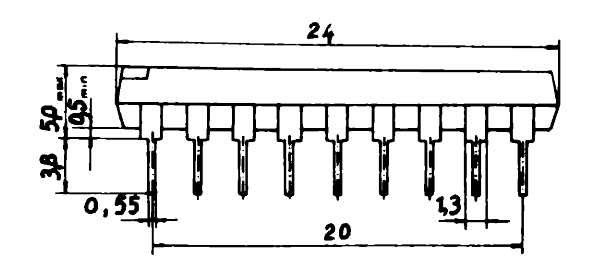


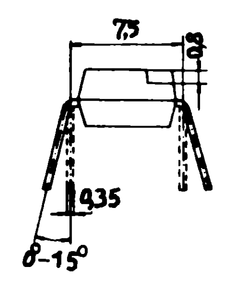
Bild 1.6: Kombinierte Ein- und Ausgangsschaltung des U 808 D

Die Umschaltung zwischen Eingangs- und Ausgangsverhalten erfolgt intern durch die Zeitsteuerung und Befehlsdekodierung der ZVE.
In den Zuständen T1, T2, T4 und T5 werden die Stufen immer auf Ausgang geschaltet. Im Zustand T3 werden die Stufen in Abhängigkeit vom gerade in Abarbeitung befindlichen Befehl und in Abhängigkeit vom Zyklus entweder auf Ausgang oder auf Eingang geschaltet.
In den Zuständen WAIT und STOP sind die Ausgangstransistoren hochohmig, die Ausgangsstufen sind "schwimmend" (floating).
Der eingangsseitige Transistor wird dabei nicht abgeschaltet.

1.5.4. Konstruktive und elektrische Kennwerte

Die ZVE U 808 D befindet sich in einem 18poligen DIL-Plastgehäuse mit 7,5 mm Reihenabstand und einem Rastermaß von 2,5 mm.





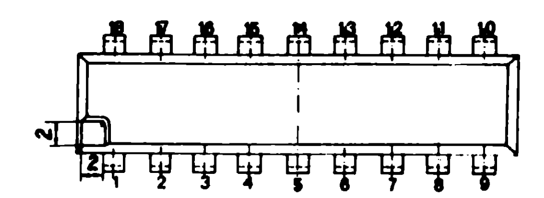


Bild 1.7: Gehäuseabmessungen des U 808 D

Die Gehäusemaße und die Anschlußbelegung zeigen Bild 1.7 und Tafel 1.6. In den Tafeln 1.7 und 1.8 sind die elektrischen Kennwerte des U 808 D zusammengefaßt. Die Bilder 1.8 und 1.9 zeigen die zugehörigen Impulsdiagramme.

Tafel 1.6: Anschlußbelegung des U 808 D

Pin	Kurzbezeichnung	Erläuterung
1	U_{DD}	Betriebsspannung U_{DD}
2	D7	Datenbus-Ein/Ausgang 7
3	D6	" " " 6
4	D5	" " " 5
5	D4	" " " 4
6	D3	" " " 3
7	D2	" " " 2
8	D1	" " " 1
9	D0	" " " 0
10	U_{CC}	Betriebsspannung U_{CC} (Bulk)
11	S2	Status-Ausgang 2
12	S1	" " 1
13	S0	" " 0
14	SY	Synchronisations-Ausgang
15	C2	Takteingang 2
16	C1	" 1
17	RD	READY-Eingang
18	IT	INTERRUPT-Eingang

Tafel 1.7: Grenzwerte und statische Kennwerte des U 808 D

<u>Grenzwerte</u> (Spannungen bezogen auf $U_{CC} = 0$)

Betriebsspannung	U_{DD}	-20 V...+0,3 V
Eingangsspannungen	U_e	-20 V...+0,3 V
Betriebstemperatur	ϑ_a	0...+70 °C
Lagertemperatur	ϑ_s	-55 °C...+125 °C
Verlustleistung	(ϑ_a = 25 °C)	1 W

<u>Statische Kennwerte</u>

			min.	typ	max.	Einheit
Betriebsspannung	U_{DD}		-9,45	-9	-8,55	V
Betriebsspannung	U_{CC}		4,75	5	5,25	V
Eingangsspannung	U_{eH}	U_{CC} -1,5			U_{CC}+0,3	V

- -

		min.	max.	
Stromaufnahme bei I_{aL} = 0,4 mA U_{DD} = -9,45 V; U_{CC} = +5,25 V	$-I_{DD}$		60	mA
Eingangsreststrom bei U_e = 0 V; U_{DD} = -9,45 V; U_{CC} +5,25 V	$/I_e/$		10	µA
Ausgangsspannungen bei R_L = 12 kΩ , C_L = 100 pF	U_{aL}		0,4	V
bei R_L = 12 kΩ , C_L = 100 pF	U_{aH}	2,4		V
Eingangskapazitäten	C_e		10	pF
Ausgangskapazitäten	C_a		10	pF

Tafel 1.8: Dynamische Kennwerte des U 808 D

<u>Dynamische Kennwerte</u>

		min.	max.	Einheit
Taktperiode	T_{zy}	2	3	µs
Eingangsanstiegzeit	$t_{HL} = t_{LH}$		50	ns
Impulslänge	t_{C1HH}	600		ns
Impulslänge	t_{C2HH}	550		ns
Taktabstand	t_{C1C2LL}	900	1100	ns
Taktabstand	t_{C2C1}	400		ns
Taktabstand	t_{C1C2}	200		ns
Verzögerungszeit SY	t_{vsy}		700 [1]	ns
Verzögerungszeit S0, S1, S2	t_{vs}		1100 [1]	ns
Verzögerungszeit S2 bei Interrupt	t_{vs2I}		1000 [1]	ns
Verzögerungszeit D0...D7	t_{VD}		1000 [1]	ns
Haltezeit D0...D7	t_{HD}	100 [1]		ns
Bereitstellzeit D0...D7	t_{BD}	0		ns
Nachwirkzeit D0...D7	t_{ND}	t_{vsy}		ns
Bereitstellzeit RD = 1	$t_{BR\ 1}$	350		ns
Bereitstellzeit RD = 0	t_{BR0}	0		ns
Nachwirkzeit RD	t_{NR}	200		ns
Bereitstellzeit IT = 1	t_{BI1}	200		ns
Bereitstellzeit IT = 0	t_{BI0}	0		ns
Nachwirkzeit IT	t_{NI}	200		ns

[1] bei $U_{DD} = -8{,}55$ V; $U_{CC} = 4{,}75$ V; $U_{eL} = 0{,}4$ V; $U_{eH} = 3{,}25$ V;
$R_L = 12$ kΩ ; $C_L = 100$ pF

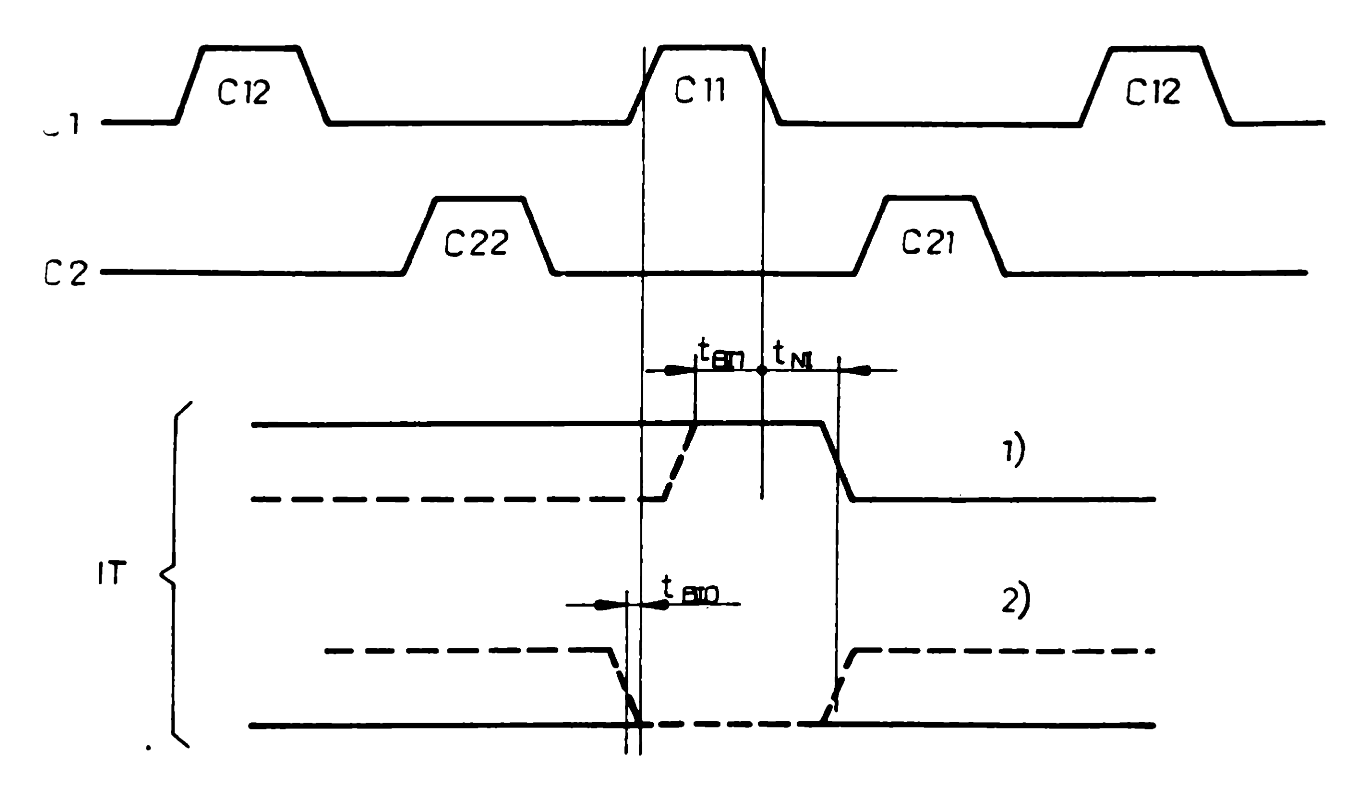


Bild 1.8: Zeitbeziehung des Interrupt-Signals

Die anzustrebende Betriebsausfallrate, bezogen auf den durch den Schaltkreis verursachten Funktionsausfall von Geräten und Anlagen beträgt bei einer angestrebten Betriebsdauer von mindestens 1000 h gemittelt über jeweils 12 Monate bei Betrieb mit mittlerer elektrischer Belastung, bei einer mittleren Umgebungstemperatur von 45 °C und vernachlässigbarer mechanischer Belastung

$$\lambda_B \leq 1 \cdot 10^{-5}\ h^{-1}.$$

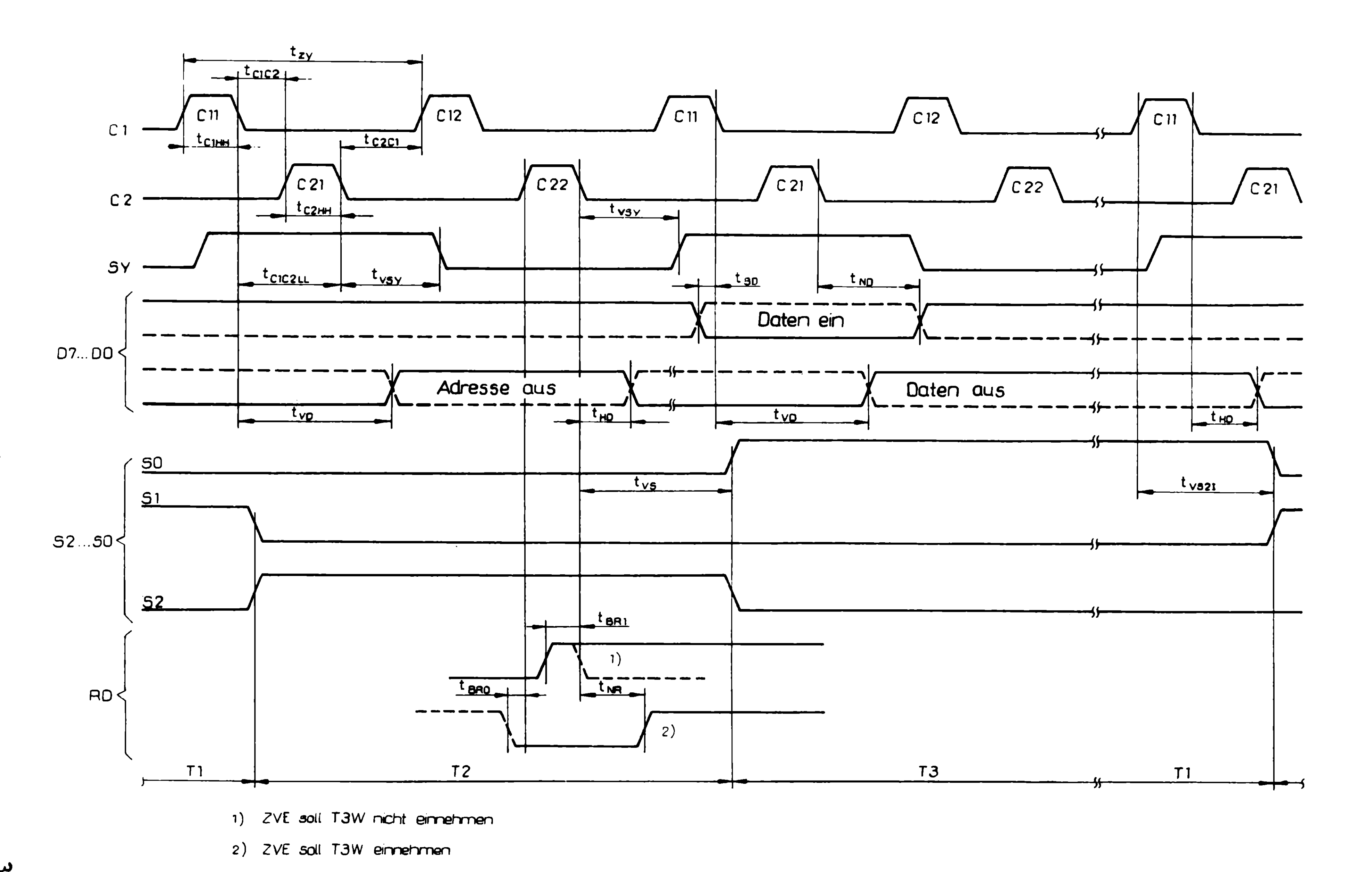


Bild 1.9: Impulsdiagramm zur Definition der dynamischen Kennwerte

2. Hardware - Beschreibung des Mikrorechners MRS1

2.1. Prinzipielle Struktur eines Mikrorechners

Der Einsatz von Mikroprozessoren erfolgt immer in ausgebauter Form, dem Mikrorechner (µR). Zu einem solchen Rechner gehören i.A. folgende Baugruppen:

- Prozessor-Baugruppe (CPU-Baugruppe)
- Programmspeicher (ROM-Baugruppe)
- Datenspeicher (RAM-Baugruppe)
- Ein-/Ausgabebaugruppe (IN/OUT)
- Zusatzlogik.

Die Prozessorbaugruppe stellt das eigentliche "Herz" des Rechners dar. Sie dient zur Steuerung und Ausführung aller mit dem µR ausführbaren Datenmanipulationen, ihre technischen Daten (z.B. Befehlsumfang, Zykluszeiten) bestimmen wesentlich die Leistungsfähigkeit des gesamten Rechners. Der Programmspeicher enthält das Steuerprogramm (die sogenannte Software) des µR, während der Datenspeicher zur Abspeicherung von Informationen, die sich während des Betriebs ändern, dient. Bei Stromabschaltung verliert der Datenspeicher im Gegensatz zum Programmspeicher seinen Inhalt.
Die Ein-/Ausgabebaugruppe dient zur Kommunikation des Rechners mit externen Geräten.
Die Baugruppe Zusatzlogik erweitert die Anwendungsmöglichkeiten des Rechners, über ihre Steuereingänge kann die Betriebsweise des µR beeinflußt werden.

2.2. Beschreibung des Systemaufbaus des MRS1

2.2.1. Blockaufbau des Mikrorechners

Damit der Rechner MRS1 aufwandsoptimal an die verschiedensten Anwendungsfälle angepaßt werden kann, wurden alle Baugruppen so weit wie möglich modular konzipiert. Das bedeutet, daß entsprechend den Einsatzanforderungen eine Abrüstung des Systems in weiten Grenzen möglich ist.

Der µR besteht aus folgenden Baugruppen:

- Prozessorbaugruppe (Leiterkarten CPU 1 und CPU 2)
- Speicherbaugruppe (max. 8 Leiterkarten RAM/ROM)
- Ein-/Ausgabebaugruppe
  (max. 2 Leiterkarten IN/OUT, max. 2 Leiterkarten OUT)
- Interruptbaugruppe
  (1 Leiterkarte INT mit max. 8 Funktionseinheiten)

Hinzu kommt die AD-Baugruppe, die eine Analog-Digital- bzw. Digital-Analog-Umsetzung vornimmt, und zur Verarbeitung von analogen Systemvariablen dient. Ein konstruktiv auf das System abgestimmtes Netzteil (NT) erzeugt die notwendigen Versorgungsspannungen. Die Minimalkonfiguration des Rechnersystems besteht aus 4 Leiterkarten:

- 2 Leiterkarten Mikroprozessor (CPU 1, CPU 2)
- 1 Speicher-Karte (RAM/ROM) mit einer maximalen Kapazität von 1,75 kbyte ROM (1 k = 1024) und 0,25 kbyte RAM - Speicherplatz
- 1 Ein-/Ausgabe-Karte (IN/OUT) mit einer maximalen Kapazität von 4 Eingabe- und 4 Ausgabetoren mit je 8 bit Breite.

Die Baugruppen sind in den Abschnitten 2.4. bis 2.9. detailliert beschrieben.

2.2.2. Periphere Anschluß- und Steuermöglichkeiten.

Der MRS1 besitzt maximal 64 digitale Ein- und 192 digitale Ausgänge, die in Toren zu je 8 bit Breite zusammengefaßt sind. Die Tore innerhalb der IN/OUT- bzw. OUT-Baugruppen dienen dem Datentransfer. Extern bestehen zusätzliche Erweiterungsmöglichkeiten, wenn entsprechende Verzweigungen vorgesehen werden.
Durch die AD-Baugruppe besitzt der Rechner dazu auch analoge Ein- und Ausgänge, deren Anzahl von der jeweiligen Konfiguration des Wandlers abhängt.
Mittels der Steuereingänge INT REQ bzw. HLT INT REQ, WAIT REQ und HOLD REQ kann Einfluß auf die Betriebsweise bzw. -art des µR genommen werden.

2.2.3. Bauelementeeinsatz

Die schaltungstechnische Konzeption des Rechnersystems beruht auf der ausschließlichen Verwendung von im RGW in Produktion bzw. Entwicklung befindlichen Bauelementen.
Hierzu gehören auch hochintegrierte MOS-LSI-Schaltkreise (U 808 D, U 501 D, CM 8001), die im Prozessor und der Speicherbaugruppe eingesetzt werden. Die Steuer- und Zusatzelektronik aller Baugruppen wurde mit TTL-Bauelementen der D 10- und D 20- (D 204) Serien aufgebaut.

2.2.4. Konstruktiver Aufbau

Aufgrund des Haupteinsatzgebietes des Mikrorechners, der BMSR-Technik, wurde bei der konstruktiven Gestaltung das EGS-Gefäßsystem zugrunde gelegt.
Alle Leiterkarten haben die Abmessungen 215 x 170 mm, sie sind mit zwei 58poligen direkten Steckverbindern ausgerüstet. Der Schwierigkeitsgrad der doppelseitig gedruckten Leiterkarten mit Durchkontaktierungen beträgt 3 (bzw. bei RAM/ROM-Karte Schwierigkeitsgrad 4). Das Netzteil ist konstruktiv auf das System abgestimmt, es ist als Einschub mit den Abmessungen 215 x 170 x 160 mm ausgeführt.
Bei Aufbau des Mikrorechners wird als Gefäß eine EGS-Doppelkassette (EGS-Baugruppeneinsatz für Karteneinschub 480 x 240 x 240) empfohlen.

2.3. Anwendungs- und Erweiterungsmöglichkeiten des Mikrorechnersystems

2.3.1. Einsatzmöglichkeiten des µR MRS1

Aufgrund der Konzeption des µR, analoge und digitale Ein- und Ausgänge, modularer Aufbau und ausschließlicher Einsatz von RGW-Bauelementen, stellt er eine für viele Anwendungsfälle günstige Systemlösung dar. Der Einsatz des Rechners ist vor allem für Regelungs- und Steuerungsaufgaben gedacht, wie sie z.B. in der BMSR-Technik auftreten.
Der Rechner hat eine typische Befehlsausführungszeit von 20 µs, das bedeutet eine gewisse Einschränkung für zeitkritischen Echtzeitbetrieb.

2.3.2. Erweiterungsmöglichkeiten der Betriebsarten

Durch entsprechende externe Steuermaßnahmen kann der µR die im folgenden beschriebenen Betriebsarten einnehmen:

2.3.2.1. DMA-Betrieb

Der Datenaustausch zwischen Rechner und externen Geräten erfolgt i.A. über die IN/OUT-Baugruppe. Ist eine sehr große Anzahl von Daten zu übertragen, erfordert das bei Verwendung dieser Baugruppe sehr viel Rechenzeit. Um diese Problematik zu umgehen, können die Daten auch direkt in den Datenspeicher ein- bzw. ausgelesen werden. Das erfolgt im sogenannten DMA-Betrieb (DMA = Direkt Memory Access, Direkter Speicherzugriff).
Hierzu muß ein Sender mit tri-state-Ausgängen an die Speicherplatzadressierung ($\overline{\text{MAD 0}}$ - $\overline{\text{MAD 13}}$) und den Datenbus (DB) angeschlossen werden. Für diese Sender wird der Einsatz von Schaltkreisen D 103 vorgeschlagen, die im normalen Rechenbetrieb hochohmig geschaltet werden. Über diese Schaltkreise dürfen nur Daten in den Rechner eingespeist werden, wenn der Prozessor eine Rückmeldung der DMA-Betriebsweise mittels des Signals $\overline{\text{HOLD ACK}}$ vorgenommen hat.
Um einen DMA-Betrieb durchführen zu können, muß der Prozessor erst in einen der Zustände T3S oder T3W kommen, z.B. mittels einer WAIT-Anforderung ($\overline{\text{WAIT REQ}}$), bevor die Übernahmeanforderung ($\overline{\text{HOLD REQ}}$) wirksam wird.
Der Datenaustausch kann nun erfolgen, indem über die Senderbausteine eine Speicherzelle adressiert wird und dann die Dateninformation ausgetauscht wird. Hierbei ist die Zeitsteuerung extern vorzunehmen.
Verläßt der Prozessor den Zustand T3W bzw. T3S, wird die DMA-Betriebsart sofort abgebrochen.

2.3.2.2. Synchronisation zu externen Geräten

Um den Mikrorechner mit dem Betrieb externer Geräte zu synchronisieren, existieren verschiedene Möglichkeiten. Mittels der Interrupt-Baugruppe können Unterprogrammroutinen aufgerufen werden, die dann den Datentransport mit dem externen Gerät ausführen.

Eine weitere Möglichkeit wäre das software-mäßige zyklische Abfragen der Eingangstore des Rechners (Polling).
Außerdem kann der Rechner durch das Steuersignal $\overline{\text{WAIT REQ}}$ in einen Warte-Zustand gebracht werden, bis das entsprechende periphere Gerät wieder Daten benötigt oder sendet.

2.3.2.3. Schritt-Betrieb

Der Schritt-Betrieb (SST-Single Step) dient grundsätzlich für Prüfzwecke. Bild A 2.1 zeigt die Schaltung einer extern an den Rechner anzuschließenden Baugruppe, die diese Betriebsart realisiert.
Der Prozessor befindet sich im Zustand T3W (WAIT). Mittels Tastendrucks wird er für genau einen Maschinenzyklus freigegeben, er nimmt dann wieder den Zustand T3W ein. Es kann somit die Funktion des Mikrorechners schrittweise überprüft werden. Günstigerweise bietet sich die Beobachtung und Auswertung der Adressen MAD'0...MAD'15 an, deren Logikzustände beispielsweise über Treiberstufen mittels LED's (VQA 12) angezeigt werden können.

2.4. Prozessorbaugruppe

2.4.1. Aufbau der CPU-Baugruppe

Die CPU-Baugruppe stellt die Steuereinheit des gesamten µR dar. Sie besteht aus den Leiterkarten CPU 1 und CPU 2. Der CPU-Schaltkreis U 808 D befindet sich auf der Karte CPU 1, außerdem wird auf dieser Platte die Verteilung und Aufspaltung sämtlicher Daten- und Adreßinformationen vorgenommen. Auf der Leiterkarte 2 werden alle Takt- und Steuersignale, sowohl für die CPU-IS, als auch zur Ansteuerung aller anderen Baugruppen des µR, erzeugt.

2.4.2. Funktionseinheiten der CPU-Baugruppe

Die CPU-Baugruppe besteht aus den nachfolgend kurz beschriebenen Baueinheiten:

2.4.2.1. CPU-Schaltkreis

Die CPU-IS U 808 D stellt den Kern der CPU-Steuerung dar, sie bestimmt wesentlich die Arbeitsweise der CPU und die Struktur der externen Elektronik.

Die Datenkommunikation erfolgt über einen 8-bit-Datenbus (D0 - D7). Hierbei werden die Steuerinformationen (MAD 14,15), 14 Adreßbits und Daten zeitmultiplex zwischen der CPU-IS und den Datentreibern ausgetauscht. Die Eingänge RDY und INT werden zur Steuerung der IS genutzt; die Ausgänge S0, S1 und S2 in Verbindung mit SY signalisieren den jeweiligen Zustand der IS U 808 D.

2.4.2.2. Taktgenerator

Der Taktgenerator (Clock Generator) erzeugt die beiden zum Betrieb der IS U 808 D notwendigen, ineinander verschachtelten Impulsfolgen cp 1 und cp 2. Diese Taktimpulse werden durch eine entsprechende sequentielle Logik aus einer quarzstabilisierten Taktfolge abgeleitet. Für den Quarzoszillator wurde eine kapazitive Dreipunktschaltung gewählt, die eine entsprechende spannungs- und temperaturmäßige Konstanz aufweist. Das Impulsdiagramm des Taktgenerators ist im Bild A 2.2 dargestellt.

2.4.2.3. Hilfszeitbildung (Aux. Timing)

In dieser Funktionseinheit werden wichtige Steuer- und Taktsignale, die grundsätzlich immer zum Betrieb der CPU-Baugruppe notwendig sind, erzeugt. Aus dem Ausgangssignal sync' der CPU-IS wird der Synchronisierimpuls Sync A durch Bezugnahme auf die Taktimpulsflanken gewonnen. Mit diesem Impuls erfolgt die Auswahl des Taktimpulses cp 12. Alle anderen in dieser Einheit erzeugten Signale, dienen zur Steuerung des Datentransports. (siehe auch Impulsdiagramm Bild A 2.2.)

2.4.2.4. Statusdekoder

Der Statusdekoder (State Decoder) dient zur Dekodierung des jeweiligen Zustandes der CPU-Baugruppe aus den von der IS angebotenen Statusausgängen S0, S1 und S2. Als zeitliche Referenz wird der Taktimpuls cp 12 benutzt.
Entsprechend der internen Logik der IS U 808 D werden während der Zustände T1 bzw. (T1I) und T2 ~~Daten (bzw.~~ Adressen) von der CPU ausgesendet, während T3 übernimmt der Prozessor die auf dem Datenbus liegenden Daten. Um hierfür eine Vorbereitung zu treffen, bzw. um Einschwingvorgänge unwirksam zu machen, ist ein Steuer-

signal T3A notwendig, das den Zustand T3 überlappt. Dieses Signal wird in der oben beschriebenen Funktionsgruppe Aux Timing erzeugt.
Die Signale T3S (Stop) und T3W (Wait) sind ebenfalls mit cp 12 verknüpft, sie können aber mehrere Zustände lang dekodiert sein. Die Zustände T4 und T5 werden nicht für Steuerzwecke benötigt, sie werden deshalb nicht extra dekodiert (siehe auch Bild A 2.2.).

2.4.2.5. Zyklusdekoder (Cycle Decoder)

Die Zyklusdekodierung dient zur externen Kennzeichnung des gerade in Abarbeitung befindlichen Zyklus in der CPU. Die Ausführung von Befehlen erfolgt je nach Befehlsform in einem, zwei oder drei Zyklen. Der erste Zyklus ist hierbei immer ein Befehlsholzyklus (PCI). Die anderen Zyklen können Datenschreib- (PCW), Datenlese- (PCR) oder Ein-/Ausgabezyklen (PCC) sein. Die Zykluskodierung wird in jedem Zyklus während T2 auf den beiden höchstwertigen Adreßbits (MAD 14, 15) ausgegeben.
In der Funktionseinheit werden diese Informationen auch noch mit Statussignalen zu Datenübernahmebefehlen verknüpft.

2.4.2.6. Eingangsmultiplexer

In dieser Funktionsgruppe (Input Multiplexer) erfolgt eine zeitmultiplexe Zusammenfassung der von der Speicherbaugruppe (MDB-Bus) der I/O-Baugruppe (INB-Bus) und der von der Interruptkarte (IIB-Bus) kommenden Dateninformationen zum Eingangsdatenbus (DI) der CPU. Die zeitliche Auswahl erfolgt hierbei über einen 1-aus-3-Dekoder.
Die Ansteuerung der Datenbusse MDB und INB erfolgt über parallel geschaltete open-collector-IS D103 der Leiterkarten RAM/ROM bzw. der Leiterkarten IN/OUT. Der Datenbus IIB hat aufgrund des Formates des Befehls RST (Restart) folgende Konfiguration:

0	0	IIB 5	IIB 4	IIB 3	1	0	1

Es brauchen also nur drei Dateninformationen (IIB 3, 4 und 5) übertragen werden.

2.4.2.7. Dateneingangstreiber

Der Dateneingangstreiber (Data Bus In Driver) dient zur Übernahme der Eingangsdateninformationen (DI) auf den Datenbus (D) der CPU-IS.
In der Funktionsgruppe wird hierzu eine zeitliche Auswahl mit dem Signal $\overline{\text{DB IN}}$ getroffen, um eingangsseitig dem tristate-Verhalten des CPU-Datenbusses (D) zu entsprechen.

2.4.2.8. Datenausgangstreiber

Der Datenausgangstreiber (Data Bus Out Driver) dient zur pegel- und strommäßigen Anpassung des CPU-Datenbusses (DO) an den Datenbus (DB) des Mikrorechners. Mit dem Signal $\overline{\text{DB OUT}}$ kann in bestimmten Betriebsarten des µR (z.B. DMA-Betrieb) dieser Informationsfluß unterbrochen werden.
Die Transistoren T 101-108 stellen eine günstige Alternative zum Einsatz von Import-Low-Power-TTL-IS an den Ausgängen der IS U 808 D dar.

2.4.2.9. Adreßzwischenspeicher

Diese Funktionseinheit (Address Latch) dient zum Auffangen der während der Zustände T1 und T2 von der CPU ausgesendeten Adreßinformationen und der Zykluskennung in entsprechenden Zwischenspeichern.
Bei speicherbezogenen Befehlszyklen (PCI, PCR, PCW) erfolgt eine Abspeicherung von 14 Adreßbits (MAD 0 - MAD 13), die zur Selektion des jeweiligen Speicherplatzes dienen; bei IN/OUT-Befehlszyklen (PCC) erfolgt während T1 eine Zwischenspeicherung der Ausgabedaten (MAD 0 - 7) und während T2 die Speicherung des Kodewortes (MAD 9 - 13) zur Adressierung des ausgewählten IN/OUT-Ports.
Die ebenfalls gespeicherten Informationen MAD 14, 15 dienen zur Zykluskennung.
Die direkten Ausgänge der Funktionsgruppe (MAD'0 - MAD'13) dienen zur Ansteuerung der IN/OUT- und OUT-Baugruppen, die Ausgänge MAD'14 und MAD'15 steuern den Zyklusdekoder.
Die Speicher-Baugruppen (RAM/ROM) werden über die negierten Ausgänge ($\overline{\text{MAD 0}}$ - $\overline{\text{MAD 13}}$) adressiert. Diese Ausgänge können mit dem Signal $\overline{\text{HOLD ACK}}$ abgeschaltet werden (bei DMA-Betrieb).

In dieser Betriebsweise kann die Speicherplatzadressierung dann über parallel zu schaltende Schaltkreise D 103 (mit offenem Kollektor) erfolgen, hierbei ist der Anschluß eines Gatterausgangs pro Bit zugelassen.

2.4.2.10. Interrupt-Logik

Im Mikrorechner MRS 1 kann eine Interruptbearbeitung nur mit Hilfe des RESTART-Befehls (RST) erfolgen. Deshalb sind acht unterschiedliche Interruptanmeldungen möglich, die Kodierung dieser Unterbrechungen werden dem Prozessor über den Interrupt-Datenbus (IIB 3, 4, 5) zugeführt.
Durch die Funktionseinheit Interrupt-Logik (Interrupt Logic) kann eine von außen (i.A. von der Interrupt-Baugruppe) kommende Unterbrechungsanforderung über die Eingänge $\overline{\text{INT REQ}}$ oder $\overline{\text{HLT INT REQ}}$ bearbeitet werden. Eine Interruptanforderung über den Eingang $\overline{\text{HLT INT REQ}}$ löst nur dann das Setzen des Interrupt-Einganges der CPU-IS (IT) aus, wenn sich der Prozessor im Zustand T3S (STOP) befindet. Im Gegensatz dazu ist mit Hilfe des Signals $\overline{\text{INT REQ}}$ eine Unterbrechung während beliebiger Zustände (außer T3W) möglich; die laufende Programmabarbeitung kann also hiermit unterbrochen werden. In diesem Fall arbeitet die CPU den gerade anliegenden Befehl zu Ende und nimmt im nächsten Befehlsholzyklus (PCI) anstelle T1 den Zustand T1I ein.
Die Interrupt-Logik setzt daraufhin den IT-Eingang der CPU-IS zurück und schaltet den Eingangsmultiplexer mit dem Signal $\overline{\text{INT SEL}}$ auf den Interruptdatenbus (IIB). Über diesen Datenbus erhält die CPU dann während T3 die Kodierung des Restart-Befehls (RST).
Das Ausgangssignal $\overline{\text{INT REQ LTH}}$ dient als Rückmeldung für die Unterbrechungsanforderung und -bearbeitung durch die CPU.

2.4.2.11. Übernahmelogik

Die Betriebsweise "Direkter Speicherzugriff" (DMA, Direct Memory Access) wird mittels der Funktionseinheit Übernahmelogik (Hold Logic) gesteuert. DMA-Operationen können nur während der Prozessorzustände T3W (WAIT) oder T3S (STOP) durchgeführt werden, deshalb muß im normalen Rechenbetrieb (Prozessor nicht in T3S)

der DMA-Anforderung (HOLD REQ) eine WAIT-Anforderung (WAIT REQ) vorausgehen. Durch die Funktionsgruppe erfolgt nach Übernahme der HOLD-Anforderung die Abschaltung des Prozessordatenbus DB und der Speicherplatzadressierung $\overline{\text{MAD}}$ durch die Signale $\overline{\text{HOLD ACK}}$ und $\overline{\text{DB OUT}}$.
Diese Abschaltung der Datenbusse wird aufgehoben, wenn die DMA-Anforderung entfällt (die CPU bleibt im Zustand T3W bzw. T3S), wenn die WAIT-Anforderung entfällt (die CPU setzt die Programmabarbeitung mit dem Zustand T3 fort) oder wenn eine Interruptanforderung erfolgt ist (Prozessor beginnt mit der Interruptbearbeitung durch Zustand T1I bzw. bleibt bei bestehender WAIT-Anforderung im T3W-Zustand).
Das Ausgangssignal $\overline{\text{HOLD ACK}}$ dient außerdem zur Rückmeldung der DMA-Betriebsweise nach außen.

2.4.2.12. Anlauflogik

Diese Funktionseinheit dient zur Steuerung des Prozessoranlaufs und der Betriebsspannungsüberwachung. Außerdem ermöglichen die Signale $\overline{\text{CLE}}$ und $\overline{\text{CI}}$ den Start der Programmbearbeitung bzw. zyklische Interruptanforderungen. Der zeitliche Abstand zwischen Interruptimpulsen beträgt 40 ms und ist für viele Anwendungsfälle geeignet. Unterschreitet die Betriebsspannung den Schwellwert von U_{CC} = 4.75 V, erfolgt eine Rücksetzung durch das Löschsignal ($\overline{\text{CLA}}$ = Low); hierdurch wird der Taktgenerator gesperrt, Interruptanmeldungen an die CPU werden unwirksam und die Anlauflogik wird in den Ausgangszustand zurückgesetzt.
Bei Benutzung der Interruptausgänge ($\overline{\text{CLE}}$, $\overline{\text{CI}}$) sind diese mit den entsprechenden Eingängen ($\overline{\text{IA}}$ 0, 1) der Interruptkarte zu verbinden.

2.4.3. Einsatz der CPU-Baugruppe

Die Prozessoreinheit ist durch die Zusammenschaltung der Leiterkarten CPU 1 und CPU 2 in sich funktionsfähig. Zur Stromversorgung sind folgende Betriebsspannungen erforderlich:

CPU 1:

$U_{CC} = 5\ V \pm 5\ \%$ $(I_{CCtyp} = 400\ mA)$

$U_{DD} = -9\ V \pm 5\ \%$ $(I_{DDtyp} = 30\ mA)$

CPU 2:

$U_{CC} = 5\ V \pm 5\ \%$ $(I_{CCtyp} = 600\ mA)$

$U_3 = 15\ V \pm 5\ \%$ $(I_{3typ} = 30\ mA)$

$U_4 = -15\ V \pm 5\ \%$ $(I_{4typ} = 20\ mA)$

$U_5 = 1\ V$, 50 Hz (Taktreferenzsignal)

Die Anschlußbelegungen der CPU-Leiterkarten sind in Tafel A 2.15.1. und A 2.15.2. angegeben.

2.5. RAM/ROM-Baugruppe

2.5.1. Funktion und allgemeine Beschreibung

Die RAM/ROM-Baugruppe ist die Speicherbaugruppe des MRS 1. Die ROM's haben die Funktion des Programmspeichers, die RAM's stellen den Datenspeicher dar. Wie schon im Punkt 2.1. erwähnt, ist damit in den ROM's das Steuerprogramm (Software) des µR abgespeichert, die RAM's dienen der Abspeicherung von Daten, die sich während des Rechner-Betriebs ändern.
Die gesamte direkt adressierbare Speicherkapazität beläuft sich auf 16 k Worte (1 Wort = 8 bit = 1 byte).
Die hier verwendete Konzeption der RAM/ROM-Karte des MRS 1 mit RAM- und ROM-Bauelementen auf einer Platte stellt eine von vielen möglichen Varianten der Speicherbaugruppe dar.
Sie verfügt über eine maximale Speicherkapazität von 2 k Worten. Davon sind 1,75 k Worte als ROM-Kapazität vorhanden, 0,25 k Worte frei wählbar als RAM- oder ROM-Kapazität.
Da für alle (im Maximalfall 8) RAM/ROM-Karten 1 Leiterkartentyp Verwendung findet, ergeben sich einerseits Möglichkeiten der reduzierten Kartenbestückung, andererseits sind aber Besonderheiten bezüglich Lastfaktoren zu beachten.

2.5.2. Organisation der Speicherblöcke

Die im folgenden als Speicherblöcke bezeichneten Einheiten haben jeweils eine Kapazität von 0,25 k byte.
Jeder Speicherblock besteht aus einem (P)ROM mit der Organisation 256 x 8 z.B. U 501 D, 1602 A, 1702 A usw.; bzw. aus 8 RAM-Bauelementen der Organisation 256 x 1, z.B. CM 8001, 1101 A usw.
Die Auswahl jeweils eines dieser Blöcke erfolgt mittels der Adressen MAD 8...MAD 13, die von der CPU 1-Karte zur Verfügung gestellt werden. Die Adressen MAD Ø...MAD 7 dienen zur Auswahl eines 8 bit-Wortes innerhalb eines ROM- bzw. RAM/ROM-Blocks.
Die Organisation der gesamten Speicheranordnung bei voller Speicherkapazität ist im Bild A 2.4. gezeigt.

Beispiel:

Ist nach der Adressierung des mit Pfeil versehenen RAM/ROM-Blocks gefragt, so ergibt sich diese aus den eingezeichneten Adressen,

der DMA-Anforderung (HOLD REQ) eine WAIT-Anforderung (WAIT REQ) vorausgehen. Durch die Funktionsgruppe erfolgt nach Übernahme der HOLD-Anforderung die Abschaltung des Prozessordatenbus DB und der Speicherplatzadressierung $\overline{\text{MAD}}$ durch die Signale $\overline{\text{HOLD ACK}}$ und $\overline{\text{DB OUT}}$.
Diese Abschaltung der Datenbusse wird aufgehoben, wenn die DMA-Anforderung entfällt (die CPU bleibt im Zustand T3W bzw. T3S), wenn die WAIT-Anforderung entfällt (die CPU setzt die Programmabarbeitung mit dem Zustand T3 fort) oder wenn eine Interruptanforderung erfolgt ist (Prozessor beginnt mit der Interruptbearbeitung durch Zustand T1I bzw. bleibt bei bestehender WAIT-Anforderung im T3W-Zustand).
Das Ausgangssignal $\overline{\text{HOLD ACK}}$ dient außerdem zur Rückmeldung der DMA-Betriebsweise nach außen.

2.4.2.12. Anlauflogik

Diese Funktionseinheit dient zur Steuerung des Prozessoranlaufs und der Betriebsspannungsüberwachung. Außerdem ermöglichen die Signale $\overline{\text{CLE}}$ und $\overline{\text{CI}}$ den Start der Programmbearbeitung bzw. zyklische Interruptanforderungen. Der zeitliche Abstand zwischen Interruptimpulsen beträgt 40 ms und ist für viele Anwendungsfälle geeignet. Unterschreitet die Betriebsspannung den Schwellwert von U_{CC} = 4.75 V, erfolgt eine Rücksetzung durch das Löschsignal ($\overline{\text{CLA}}$ = Low); hierdurch wird der Taktgenerator gesperrt, Interruptanmeldungen an die CPU werden unwirksam und die Anlauflogik wird in den Ausgangszustand zurückgesetzt.
Bei Benutzung der Interruptausgänge ($\overline{\text{CLE}}$, $\overline{\text{CI}}$) sind diese mit den entsprechenden Eingängen ($\overline{\text{IA}}$ 0, 1) der Interruptkarte zu verbinden.

2.5.3.4. Speicherblockmatrix

Hier befinden sich die Speicherbauelemente ROM und RAM. Der jeweils letzte Speicherblock einer RAM/ROM-Karte (mit den Adressen MAD 10 = 1, MAD 9 = 1, MAD 8 = 1) ist wahlweise bestückbar mit einem ROM bzw. PROM oder mit einem RAM-Block; alle anderen Speicherblockplätze sind für die Bestückung mit ROM bzw. PROM vorgesehen.

2.5.3.5. Ausgabeteil

Der Ausgabeteil IS 328...331 sowie die zum Schutz der D 100-Eingänge angeordneten Clamping-Dioden SAY 32 ermöglichen die Parallelschaltung von maximal 8 RAM/ROM-Karten. Am Ausgabeteil sind Bestückungsreduzierungen nicht möglich.

2.5.4. Karten- und Zeilenidentifikation

Die hardwaremäßige Festlegung, welche Karte welchen Adreßbereich überstreicht, wird durch externe Verdrahtung am Steckverbinder getroffen. Die Belegungen dazu sind Tafel A 2.16.3.1. zu entnehmen. Die Adreßbereiche sind in Tafel A 2.12. angegeben.

2.5.5. DMA-Betrieb

Am Eingang $\overline{WS}$ können die RAM-Blöcke bei DMA-Betrieb mit Low-Potential auf Schreiben geschaltet werden.
Der Widerstand R 322 ist aufgrund der Parallelschaltung der Eingänge $\overline{WS}$ aller RAM/ROM-Karten nur auf der 1. RAM/ROM-Karte vorzusehen.

2.5.6. Impulsdiagramm

Das Impulsdiagramm der RAM/ROM-Karte ist in Bild A 2.5. dargestellt.

2.5.7. Einsatz der RAM/ROM-Karte

Die Speicherbaugruppe ist durch Zusammenschalten der im jeweiligen Anwendungsfall benötigten RAM/ROM-Karten nach den Verdrahtungsvorschriften in sich funktionsfähig. Zur Stromversorgung jeder RAM/ROM-Karte sind folgende Betriebsspannungen erforderlich:

U_{CC} = 5 V ± 5 % (I_{CCtyp} = 230 mA)

U_{DD} = -9 V ± 5 % (I_{DDtyp} = 180 mA)

Die Anschlußbelegung der RAM/ROM-Karte ist in Tafel A 2.15.3. angegeben.

2.6. Ein-/Ausgabe-Baugruppe (IN/OUT bzw. OUT)

2.6.1. Allgemeine Beschreibung

Die IN/OUT- und OUT-Baugruppe ist die Ein- und Ausgabeeinheit des MRS 1. Sie verfügt über insgesamt 8 IN-Ports (Eingabetore) und 24 OUT-Ports. Der Eingangslastfaktor ist N_I = 1 pro bit, der Ausgangslastfaktor ist N_O = 10 pro bit.
Die voll ausgebaute Variante des Ein- und Ausgabeteils besteht aus 2 IN-OUT-Karten und 2 OUT-Karten.
Die Minimal-Variante des Ein- und Ausgabeteils enthält mindestens 1 IN-OUT-Karte, da sich auf ihr die Dekoder zur Port-Auswahl befinden.
Die für die Adressierung der Ports erforderlichen Memory-Adressen sind nicht voll identisch mit den Memory-Adressen an den RAM/ROM-Karten (z.B. bei DMA-Betrieb). Deshalb erhalten sie die Bezeichnung MAD'.
Im Bild A 2.6. ist die Ansteuerung der IN- und OUT-Ports mittels der MAD' dargestellt.

Beispiel:

Ist nach der Bezeichnung des mit Pfeil gekennzeichneten IN-Ports bzw. OUT-Ports gefragt, so entspricht diese der Wertigkeit eines aus den MAD'9...13 zusammengesetzten Wortes.
Diese ist im Beispiel:

<u>IN-PORT</u>

MAD'13 = 0
MAD'12 = 0
MAD'11 = 1   00110 ≙ 6
MAD'10 = 1
MAD' 9 = 0

Bei der vorgegebenen Belegung ist also IN 6 angesprochen.

OUT-Port

MAD'13 = 1
MAD'12 = 1
MAD'11 = 1   11101 = 29
MAD'10 = 0
MAD' 9 = 1

Damit ist OUT 29 angesprochen.

2.6.2. IN/OUT-Karte

Eine IN/OUT-Karte des MRS 1 verfügt bei voller Bestückung über 4 Eingangstore und 4 Ausgangstore. Außerdem befinden sich hier die Dekoder zur Port-Auswahl und Out-Card-Auswahl.

Sie besteht aus den im folgenden beschriebenen Funktionsblöcken:

2.6.2.1. Port-Select-Decoder

Dieser 1 aus 4-Dekoder (IS 417, 418) wird durch MAD'9 und MAD'10 angesteuert.
Hier wird unter der Voraussetzung, daß die auf der Karte befindliche Brücke in Stellung 1 geklemmt ist, (siehe Bestückungsplan Bild A 2.11.4) eins der Eingangstore 0...3 (bzw. Ausgangstore 8...11) ausgewählt, im Falle der Brückenstellung 2 eines der Eingangstore 4...7 (bzw. Ausgangstore 12...15).
Die Entscheidung hierüber ist abhängig vom MAD'11 (Auswahl der Portgruppe).

2.6.2.2. OUT-Card-Select-Decoder

Der Dekoder (IS 414, 415) wird durch MAD'12 und MAD'13 angesteuert. Hier geschieht die Auswahl, an welche OUT-Karten Informationen gegeben werden können (die insgesamt 8 Ausgangstore auf den IN/OUT-Karten 1 und 2 sind unter dem Namen "OUT 0" zusammengefaßt).
Die Auswahl der OUT-Ports auf den IN/OUT-Karten erfolgt bei H-Pegel am Anschluß OCS Ø. Bei H an OCS 1 wird die OUT 1-Karte und bei H an OCS 2 die OUT 2-Karte angewählt.

2.6.2.3. IN-Port-Multiplexer

Von den 4 Eingangstoren des IN-Port-Multiplexers (IS 401...410) wird durch den Port-Select-Decoder ein Tor angewählt und gibt unter Voraussetzung entsprechender Gruppenanwahl (durch MAD'11) seine Eingangsdaten an den IN-Bus der CPU 1-Karte weiter.

2.6.2.4. Ausgabespeicher mit Ausgangstreibern

Die Ausgabespeicher (IS 421...428) übernehmen, falls das entsprechende Tor angewählt wurde, die negierte neue Information mit der L-H-Flanke des $\overline{\text{I/O-OUT}}$ Signals an ihre Ausgänge. Die als Ausgangstreiber dienenden IS 429...436 geben die nichtnegierten Informationen nach außen.

2.6.2.5. Bestückung

Wie oben erwähnt, ist in der Grundausstattung der MRS 1 mindestens die 1. IN/OUT-Karte (IN/OUT 1) erforderlich. In der 2. IN/OUT-Karte entfallen der Port-Select-Decoder und der OUT-Card-Select-Decoder. Hingegen wird hier zusätzlich die IS 416 benötigt, die auf der IN/OUT 1 nicht eingesetzt wird. Aus dem Bestückungsplan (Bild A 2.10.8.) ist sowohl die Anordnung der Brücke als auch der IS 416 ersichtlich. Werden weniger Tore benötigt, als auf der Karte möglich sind, so können bei der Bestückung die betreffenden Schaltkreise entfallen.

2.6.3. OUT-Karte

Eine OUT-Karte des MRS 1 verfügt über 8 OUT-Tore. Die 1 aus 4-Port-Auswahl wird direkt von der 1. IN/OUT-Karte übernommen. Ebenso wird der card-sel-Eingang der OUT-Karte an einen der Anschlüsse OCS 1 oder OCS 2 der 1. IN/OUT-Karte geklemmt. Damit wird definiert, ob es sich bei der betreffenden OUT-Karte um OUT 1 oder OUT 2 handelt.
Die OUT 1-Karte verfügt über die OUT-Ports 16...23, die OUT 2-Karte über die OUT-Ports 24...31 (siehe auch Bild A 2.6.).
Das in Punkt 2.6.2.4. gesagte gilt analog für die OUT-Karten. Nichtbenötigte Tore können auch hier für die Bestückung entfallen.

2.6.4. Impulsdiagramm

Das Impulsdiagramm für die Ein- und Ausgabe ist in Bild A 2.7. angegeben.

2.6.5. Einsatz der Ein-/Ausgabebaugruppe

Zur Stromversorgung der Ein-/Ausgabebaugruppe sind folgende Betriebsspannungen erforderlich:

IN/OUT-Karte

U_{CC} = 5 V ± 5 % (I_{CC} typ = 680 mA)

OUT-Karte

U_{CC} = 5 V ± 5 % (I_{CC} typ = 980 mA)

Die Anschlußbelegungen der IN/OUT- bzw. OUT-Karten sind in den Tafeln A 2.15.4. und A 2.15.5. angegeben.

2.7. Interrupt-Baugruppe

2.7.1. Allgemeine Beschreibung

Die Interrupt-Baugruppe des MRS 1 bietet die Möglichkeit, die Programmbearbeitung mittels erzwungener Restartbefehle zu unterbrechen. Die 8 Interrupts ermöglichen die Bearbeitung von maximal 8 Interrupt-Serviceroutinen (ISRs). Die Interrupt-Baugruppe des MRS 1 bietet die Möglichkeit, daß eine jeweils begonnene ISR vollständig abgearbeitet wird, bevor andere inzwischen über die Interrupteingänge angemeldete ISRs begonnen werden. (Zur Entlastung des Stack-Registers der CPU kann zwischen der Bearbeitung zweier Unterprogramme mindestens 1 Zyklus lang ins Hauptprogramm gesprungen werden.) Für die Interrupts ist eine Rangordnung vorgegeben. INT 7, der über den Karteneingang IA 7 angemeldet werden kann, ist der ranghöchste INT Ø, der über IA Ø angewählt wird, der rangniedrigste Interrupt.

2.7.2. Kartenaufbau

Eine voll bestückte INT-Karte besteht aus folgenden Funktionsblöcken, auf deren Funktion in Punkt 2.7.3. eingegangen wird.

2.7.2.1. Anmeldeteil

Dieser Teil nimmt Interruptanmeldungen entgegen und speichert sie bis zu ihrer Bearbeitung ab. Er besteht aus den Eingangsnegatoren IS 601, 602 (mit eingangsseitigen Widerständen nach U_{CC} zur Verbesserung der Störsicherheit) und den Anmeldungs-Flip-Flops, IS 611...618 System 2.
Außerdem sind die IS 619 und die IS 610, die die Interruptanmeldung an den II-Datenbus der Prozessor-Baugruppe weiterleiten bzw. die Löschung von Anmeldungen bereits begonnener Interrupts durchführen, zum Anmeldeteil zu rechnen.

2.7.2.2. Auswahlteil

Hier wird nach Übernahme aller in den Anmeldungs-Flip-Flops gespeicherten Anmeldungen in die Auswahl-Flip-Flops die ranghöchste selektiert und binär kodiert an die Ausgänge IIB 3...IIB 5 gegeben.
Die IS 611...618 System 1 fungieren als Auswahl-Flip-Flops. Die IS 604...609 mit ihren Gattern 1 und 2 dienen der Weitergabe des Auswahl-Löschsignals, die IS 622, 623 der binären Kodierung und Informationsausgabe.

2.7.2.3. Verzögerungsteil

Die IS 620 und 621 haben die Aufgabe, bei Beginn eines Interrupts die Weitergabe der Interruptanmeldungen zur CPU zu "verriegeln" und nach Erhalt eines Interrupt-Ende-Signals (IEND) noch 4 Zyklen diese Verriegelung aufrecht zu erhalten.

2.7.2.4. Steuerlogik

Die IS 625, 626 synchronisieren den Impuls-Ablauf auf der Interruptkarte mit den Statussignalen der CPU. Damit wird sichergestellt, daß die Bereitstellung des Restartbefehlskodes noch während T3 des Interrupt-Cycles erfolgt.

2.7.3. Funktionsablauf

Die Anmeldung eines Interrupts geschieht mit einem H-L-Übergang an den Eingängen IA Ø...IA 7.
Der anmeldende Impuls muß mindestens 60 ns breit sein (Zeitdauer des Low-Zustandes). Andererseits wird im Hinblick auf Störsicherheit empfohlen, die Zeitdauer des Low-Zustandes am Eingang auf max. 6 µs zu begrenzen.
Liegen mehrere Interruptanmeldungen gleichzeitig vor, so wird innerhalb desjenigen Zyklus, in dem die CPU auf den Interruptwunsch durch den Status T1I reagiert (Interrupt-Cycle, ausgedrückt durch das Signal INT SEL), die Auswahl der ranghöchsten Interruptanmeldung getroffen, wobei der im Bild A 2.8.1. dargestellte zeitliche Ablauf vorliegt.
Mit Ende des Signals $\overline{T\ 2}$ erfolgt die Übernahme der in den Anmeldungs-Flip-Flops vorliegenden Interruptanmeldungen in die Auswahl-Flip-Flops (IS 611...618 System 1). Ab Beginn des Signals $\overline{T\ 3\ A}$ erfolgt die Auswahl des ranghöchsten Interrupts, die nach maximal 350 ns abgeschlossen ist. Dabei bleibt lediglich diejenige Anmeldung in den Auswahl-Flip-Flops bestehen, die auch als ranghöchste erkannt worden war, alle anderen in die Auswahl-Flip-Flops übertragenen Anmeldungen werden während des Auswahlprozesses gelöscht.
Die übriggebliebene Information wird binär kodiert zu den Eingängen IIB 3, IIB 4 und IIB 5 der Prozessorbaugruppe weitergeleitet.
Während des Signals $\overline{T\ 3}$ wird in den Anmeldungs-Flip-Flops diejenige Anmeldung gelöscht, die zum soeben begonnenen Interrupt gehört. Alle anderen Anmeldungen bleiben unbeeinflußt.
Die Absicherung, daß eine ISR erst vollständig abgearbeitet wird, bevor die Bearbeitung eines anderen Interrupts begonnen wird, kann folgendermaßen erfolgen (siehe auch Bild A 2.8.2.):

Die hier vorgeschlagene Lösung geht davon aus, daß ein Flip-Flop (IS 620 System 1) im Interrupt-Cycle (dem ersten Zyklus der gesamten ISR) in einen Zustand gesetzt wird, der erst am Ende der ISR aufgehoben wird. Während dieses Zustandes werden Interruptanmeldungen an den Eingängen IA Ø...IA 7 zwar entgegengenommen und gespeichert, nicht aber zur Prozessorbaugruppe weitergeleitet.

Die Wiederfreigabe für die Weiterleitung von Interruptanmeldungen zur CPU wird eingeleitet durch das Umschalten des Flip-Flops mittels eines L-H-Sprunges. Dazu muß innerhalb der ISR ein zu definierendes Bit eines OUT-Ports zunächst auf L gesetzt werden. Am Ende der Interruptbehandlung wird dieser Ausgang wieder auf H geschaltet (Interrupt-Ende-Mitteilung IEND), was den erwähnten Sprung auslöst. Voraussetzung ist, daß der Eingang IEND der Interrupt-Karte an das genannte Bit eines OUT-Ports gelegt wurde.
Nach Umschalten des Bits auf H bleibt durch den 1 : 4 Teiler IS 621 die Weiterleitung der Interruptanmeldungen noch 4 Zyklen gesperrt. In dieser Zeit geschieht auch das Rücklesen der am Anfang der Interruptbehandlung geretteten Informationen in den Akku und schließlich die Rückkehr zum übergeordneten Programm. Bei Vorliegen weiterer Interruptanmeldungen würde nun die Abarbeitung des nächsten ranghöchsten Interrupts eingeleitet werden. Für die Bearbeitung eines derartigen allgemeinen Falles sind durch äußere Verdrahtung die Kartenanschlüsse INT INH an INT DEL und CLK an T 3 A zu legen.
In einem Anwendungsfall, bei dem sicher ist, daß Interruptanmeldungen in so großen Abständen erfolgen, daß keine zeitlichen Überlappungen mit noch abzuarbeitenden ISRs entstehen, ist der Anschluß INT INH fest auf H zu legen z.B. an den Kartenanschluß HIGH.
Weitere Variationsmöglichkeiten der zeitlichen Einschiebung des Interrupts ergeben sich über die Belegung des Anschlusses IEND (z.B. durch softwaremäßige Veränderung des anliegenden Signals über das OUT-Port) und der wahlweisen Benutzung des HLT INT REQ-Anschlusses an der CPU 1-Karte.

2.7.4. Bestückung

Hardwareseitig ist die Interrupt-Baugruppe des MRS 1 so ausgelegt, daß im wesentlichen nur die Bestückung für die gewünschte Anzahl von Interruptmöglichkeiten vorgenommen werden muß. Dies gilt insbesondere für die Bestückung der Bausteine IS 611...618 mit den zugehörigen IS 603...609. Dabei ist auf den Anschluß freier Eingänge über die auf der Leiterkarte für diesen Fall

vorgesehenen Widerstände an die Spannung U_{CC} zu achten.

Beispiel: Wenn in einem Anwendungsfall nur Interrupt Ø, 1 und 2 benutzt werden sollen, können die IS 612...616, IS 603...607 und IS 601 sowie die Widerstände R 601...605, 614...616, 622, 623, 624...628 und 630 entfallen.
Bei fester Verdrahtung des Einganges IIB 5 auf der CPU 1-Karte ebenso IS 622. Die übrigen Bauelemente der Interrupt-Baugruppe sind zu bestücken.

2.7.5. Impulsdiagramm

Das Impulsdiagramm für eine Interruptbearbeitung ist im Bild A 2.8. angegeben.

2.7.6. Einsatz der Interrupt-Baugruppe

Zur Stromversorgung der Interrupt-Baugruppe ist folgende Betriebsspannung erforderlich:

$$U_{CC} = 5\ V \pm 5\ \%\ (I_{CC}typ = 270\ mA)$$

Die Anschlußbelegung der INT-Karte ist in Tafel A 2.15.6. angegeben.

2.8. Netzteil

2.8.1. Beschreibung der Baugruppe

Die Baugruppe Netzteil übernimmt die Stromversorgung des gesamten Mikrorechners. Sie stellt die TTL-Versorgungsspannung (U_{CC}), die Drainversorgungsspannung für die MOS-Bauelemente (U_{DD}), zwei Operationsverstärkerbetriebsspannungen (U3, U4) und eine Referenzwechselspannung (U5) zur Verfügung. An alle Gleichspannungen werden besondere Anforderungen an Genauigkeit und Welligkeit gestellt, deshalb fanden integrierte Spannungsregler MAA 723 (Tesla) Verwendung. Mittels dieser Regler-IS erfolgt ebenfalls eine Strombegrenzung bei den Versorgungsspannungen U_{CC} und U_{DD}, die am meisten strommäßig belastet werden. Um den Spannungsabfall über die Zuleitungen bei diesen beiden Betriebsspannungen zu minimieren, müssen die Fühlausgänge ($U_{CCfühl}/U_{EEfühl}$) an der am

weitesten entfernten zu versorgenden Baugruppe (i.A. CPU 1) mit den entsprechenden Spannungspotentialen (U_{CC} bzw. U_{EE}) verbunden werden.
Bei extremen Belastungen bzw. bei Schäden am Netzteil verhindert eine "crowbar"-Sicherung die Zerstörung der übrigen Bauelemente des Mikrorechners; wenn die Spannungen U_{CC} und U_{DD} bestimmte Grenzwerte ($U_{CC} \geq 5,7$ V, $U_{DD} \leq -9,7$ V) überschreiten, erfolgt ein ausgangsseitiger Kurzschluß dieser Spannungen mittels Thyristoren gegen Masse (U_{EE}). Hierbei wird die Sicherung Si 801 (6,3 AT) zerstört.
Beim Aufbau der Netzteilelektronik ist für die Frequenzkompensation besondere Sorgfalt zu verwenden.

2.8.2. Konstruktiver Aufbau

Um einen möglichst einfachen und unkomplizierten Aufbau zu erreichen, wurde das Netzteil als Einschub konzipiert. Die Abmessungen sind 215 x 170 x 160; der Einschub wird mit sechs Schienen im EGS-Einschubrahmen geführt. Das Netzteil besteht aus den Bauteilen Seitenplatte mit Netztransformator, Zwischenteil, Kühlplatte, Kühlkörper und Netzteilleiterkarte, die mittels Abstandsbolzen miteinander verbunden sind.
Die Wickeldaten des Netztrafos sind in Tafel A 2.14. angegeben.

2.8.3. Einsatzbedingungen

Das Netzteil erfordert eine primäre Spannungsversorgung von 220 V ± 10 % (50 Hz, 150 W). Als Versorgungsspannungen für den Rechner werden folgende Ausgangsspannungen zur Verfügung gestellt:

U_{CC} = +5 V ± 5 %, (I_{CCmax} = 7 A) (Strombegrenzung)
U_{DD} = -9 V ± 5 %, (I_{DDmax} = 2 A) (Strombegrenzung)
U_3 = +15 V ± 5 %, (I_{3max} = 1 A)
U_4 = -15 V ± 5 %, (I_{4max} = 1 A)
U_5 = 1 V ±50 %, 50 Hz (I_{5max} = 50 mA)

Bei den Versorgungsspannungen U_{CC} und U_{DD} sind die entsprechenden Fühleingänge zu beschalten.

2.9. Analog/Digital-Wandler

Der Analog/Digital-Wandler (A/D-Wandler) bildet das Bindeglied zwischen der Analogsignalquelle und dem Digitalrechner. Der nachfolgend beschriebene 10 bit-A/D-Wandler wurde dem MRS 1 zugeschnitten.
Abgeleitet aus bekannten Anwendungsfällen wurden an ihn folgende Forderungen gestellt:

Anzahl der Analogsignaleingänge:	8
Eingangsspannungsbereich:	0...+10 V Gleichspannung
Empfindlichkeit:	10 mV
Meßfehler bei ϑ_a = (25-5) °C:	0,1 % vom Bereichsende + 0,1 % vom Meßwert +1 digit
Auflösung:	1 : 1024
Betriebstemperaturbereich:	0...+70 °C

Alle digitalen Ein- und Ausgänge sollen TTL-kompatibel sein.
Der gesamte Wandler ist auf einer Steckeinheit mit den Abmessungen 170 x 215 mm unterzubringen.

2.9.1. Prinzip der A/D-Wandlung

Für den zu realisierenden A/D-Wandler wurde das Prinzip der sukzessiven Approximation verwendet.
Mit Hilfe eines Widerstandsnetzwerkes wird einem Eingang eines Komparators eine schrittweise veränderbare Vergleichsspannung zugeführt. Der Komparator vergleicht diese mit der an seinem anderen Eingang angelegten Spannung (z.B. der Spannung eines Gebers, im folgenden Eingangsspannung genannt). Ein am Komparatorausgang angekoppelter Rechner erhält die Aussage, ob die Vergleichsspannung größer oder kleiner als die Eingangsspannung ist. Der Rechner steuert das Widerstandsnetzwerk entsprechend dem eingegebenen Programm in der Form, daß nach einer minimalen Anzahl von Schritten die Vergleichsspannung den Wert der Eingangsspannung erreicht.

Die schaltungstechnische Lösung des A/D-Wandlers besteht aus den im folgenden näher erläuterten Funktionsgruppen:

- Eingangsfilter mit Spannungsteiler
- D/A-Wandler-Netzwerk
- Komparatoren mit Umschaltlogik
- Referenzquelle

2.9.2. Eingangsfilter mit Spannungsteiler

Die Eingangsfilter wurden mit den Operationsverstärkern IS 701...708 (A 109 D, siehe Stromlaufplan Bild A 2.9.7.) realisiert. Die Filter haben die Aufgabe, Störspannungsspitzen, die sich auf den Eingangssignalen befinden können, zu unterdrücken und die Gleichspannungsquellen niederohmig an die Komparatoren anzukoppeln. Die Eingangsfilter arbeiten als modifizierte Integratoren. Die Tiefpaßwirkung wird beispielsweise durch die Bauelemente R 701, P 701 und C 701 bestimmt. Für den vorliegenden Schaltungsentwurf wurde eine Zeitkonstante von 150 ms gewählt.
Da die Operationsverstärker A 109 D und die nachfolgenden Komparatoren A 110 D nur Differenzeingangsspannungen bis zu $\pm$ 5 V verarbeiten können, mußten Eingangsspannungsteiler 2,5 : 1 vorgesehen werden. Diese Teiler bestehen beispielsweise aus den Widerständen R 721 und R 701 sowie dem Dickschichtregler P 701.
Wegen der schwierigen Beschaffbarkeit von Präzisionswiderständen wurden die eingangsseitigen Teilerwiderstände durch die Reihenschaltung je eines Festwiderstandes und eines Dickschichtreglers realisiert.
Da die verfügbaren Dickschichtregler nur mit einem Temperaturkoeffizient TK $\leq 250 \cdot 10^{-6}$/grd lieferbar sind, ist ihr Widerstandswert möglichst klein zu wählen, damit ihr TK den Temperaturgang des gesamten Teilers nicht wesentlich beeinflußt. Im vorliegenden Fall wurde beispielsweise für P 701 ein Dickschichtregler von 4,7 kOhm eingesetzt, der jedoch bereits Toleranzen für die Festwiderstände R 701 und R 721 von 1 % fordert.
Ohne Veränderung der Einstellregler sind Gleichspannungsquellen mit einem Innenwiderstand bis zu ca. 60 Ohm anschließbar, wenn man einen zusätzlichen Fehler von 10 mV ausgangsseitig zuläßt.

Spannungsquellen mit höheren Innenwiderständen können durch Verringerung des Widerstandswertes dieser Dickschichtregler angepaßt werden. An dieser Stelle sei auf andere Lösungsmöglichkeiten für die Realisierung der Eingangsspannungsteiler hingewiesen:

- Zusammensetzen der Teilerwiderstände aus mehreren Festwiderständen;
- Ausmessen der geforderten Widerstände auf enge Toleranz;
- Verwendung von Dünnschicht-Widerstandsnetzwerken des VEB Keramische Werke Hermsdorf.

Besonders bei Anwendung der letztgenannten Widerstände ist eine höhere thermische Konstanz der gesamten Teilerstufe erreichbar. Die Operationsverstärker wurden in der üblichen Weise frequenzkompensiert, beispielsweise durch C 703, R 723 und C 702. Die Offsetkompensation ist ebenfalls in bekannter Weise über den nichtnegierenden Eingang der IS 701...708 beispielsweise durch P 702 möglich.

2.9.3. D/A-Wandler-Netzwerk

Das D/A-Wandler-Netzwerk dient zur Umsetzung der vom MRS 1 angebotenen Ausgabebelegung von 10 bit in einen entsprechenden Analogwert.
Die Umsetzung erfolgt über ein Widerstandsnetzwerk in Kettenschaltung. Die Zuschaltung der 10 Stufen des Netzwerkes gegen Masse bzw. gegen die Referenzspannung von 4,096 V übernehmen die Transistoren T 703...T 722.
Um die ausgangsseitige Belastung des Widerstandsnetzwerkes durch die Eingangsruheströme und Eingangsoffsetströme der 8 Komparatoren zu vermeiden, wurde ein Verstärker IS 717 mit einer Verstärkung von -1 dazwischen geschaltet. Dieser Verstärker negiert die Referenzspannung, so daß an den Komparatoreingängen zwei negative Spannungen verglichen werden. Die Offsetkompensation des Verstärkers erfolgt in der üblichen Weise mit P 717.
Das 10stufige Widerstandsnetzwerk wurde durch zwei Dünnschicht-Widerstandsnetzwerke des VEB KKWH (Erzeugnis-Nr. 4536.8-282146, Toleranz $\pm$ 0,25 %, TK $\leq 100 \cdot 10^{-6}$/grd) realisiert.

Zusammen mit dem Präzisionswiderstand R 746 (1,5 kOhm, Toleranz $\pm$ 0,25 %, TK $\leq 100 \cdot 10^{-6}$/grd) des Verstärkers wird damit die geforderte Genauigkeit und Konstanz dieses Wandlers erreicht.
Da maximal eine Vergleichsspannung von 4,092 V auftreten kann, ergibt sich durch die 10 bit Ausgabebelegung des MRS 1 eine Schrittdifferenz von

$$\frac{4{,}092\ V}{1023} = 4\ mV$$

An dieser Stelle sei erwähnt, daß dieses Netzwerk auch getrennt als D/A-Wandler Verwendung finden kann. Hierfür ist auf der Steckeinheit nur das D/A-Wandler-Netzwerk zu bestücken.

2.9.4. Komparatoren mit Umschaltlogik

Als Komparator fand der Schaltkreis A 110 D Verwendung.
Er liefert ausgangsseitig TTL-Pegel, die sofort weiter verarbeitet werden können.
Wegen der hohen Eingangsruheströme und Eingangsoffsetströme sowie der hohen Genauigkeitsforderungen konnten keine Schutzwiderstände vorgesehen werden. Damit keine Zerstörung der Schaltkreise auftritt, muß dafür gesorgt werden, daß die Spannung an den Analogsignal-Eingängen AE 1...AE 8 keinesfalls 12,5 V überschreitet.
Die Betriebsspannungsversorgung der 8 Komparatoren erfolgt über eine Z-Diode bzw. 2 Vorschaltdioden aus den Versorgungsspannungen U_4 bzw. U_3. Aus konstruktiven Gründen mußten diese Vorschaltelemente doppelt vorgesehen werden (D 703/D 706, D 704/D 707, D 705/D 708). Zur Pufferung der Versorgungsspannungen wurden die Kondensatoren C 730/C 735 und C 731/C 736 eingesetzt. Die Umschaltlogik besteht aus den Schaltkreisen IS 18 und IS 19. Mit zwei Selektionseingängen SEL 1 und SEL 2 können jeweils 4 Komparatorausgänge aktiviert werden. Im einfachsten Fall sind SEL 1 und SEL 2 an H-Potential zu legen; es ist jedoch auch eine gruppenweise Aktivierung der Ausgänge möglich. Durch Einsatz von Gattern mit open-collector-Ausgängen kann eine Erweiterung der Zahl der Eingänge erfolgen. Beispielsweise sind bei einer Erweiterung auf 16 Eingänge die entsprechenden Ausgänge zweier Steckeinheiten parallel zuschalten. Lediglich das Bitmuster an SEL 1 und SEL 2 legt dann fest, welche der Komparatorausgänge aktiviert

werden. Vorausgesetzt, daß SEL 1 auf H-Potential liegt, ergibt sich beispielsweise am Ausgang DA 1 ein H-Pegel, wenn die am zugehörigen Eingang AE 1 anliegende Spannung größer als der an den Eingängen DE 0...DE 9 liegende Binärwert in V, dividiert durch 100, ist.

2.9.5. Referenzquelle

Genauigkeit und Konstanz der Referenzspannung haben einen wesentlichen Anteil am Gesamtverhalten des A/D-Wandlers. Deshalb wurde bei der Realisierung der Referenzspannungsquelle ein relativ hoher Aufwand getrieben.
Sie besteht aus einem Referenzelement D 701, das aus einer Konstantstromquelle T 701, D 707, R 757 und R 759 mit einem Strom von 5 mA versorgt wird. Über einen nachgeschalteten Operationsverstärker IS 723 (A 109 D), dessen Verstärkung durch P 718 veränderbar ist, kann die Referenzspannung auf den gewünschten Wert von 4,096 V eingestellt werden. Der nachfolgende Transistor T 702 stellt den geforderten Ausgangsstrom für das Widerstandsnetzwerk bereit und ermöglicht gleichzeitig eine belastungsabhängige Verstärkungsregelung.

2.9.6. Abgleichhinweise

Um eine optimale Arbeitsweise des A/D-Wandlers zu erreichen, ist es erforderlich, den Abgleich der Baustufen in folgender Reihenfolge durchzuführen:

- Abgleich der Referenzspannung mit P 718 auf einen Wert von +4,096 V (M 701)
- Offsetkompensation der Eingangsfilter IS 701 bis IS 708:

  Dazu werden die Eingänge AE 1 bis AE 8 auf Massepotential gelegt (Masse des A/D-Wandlers). Mit P 702, 704, 706, 708, 710, 712, 714, 716 stellt man am Ausgang des jeweiligen Operationsverstärkers 0 V ein (M 702 bis 709).
- Offsetkompensation des Verstärkers IS 717:

  Dazu wird an DE 0 bis DE 9 ein H-Pegel angelegt und mit P 717 am Ausgang des IS 717 0 V eingestellt (M 710).

- Abgleich des Teilungsverhältnisses der Eingangsfilter:

  Dazu wird an DE 0 bis DE 9 ein L-Pegel, an die Eingänge AE 1 bis AE 8 eine Gleichspannung von +10,230 V angelegt.
  Jetzt werden die Regler P 701, 703, 705, 707, 709, 711, 713, 715 so verändert, daß der zugehörige Komparatorausgang gerade von L nach H kippt (M 711 bis 718).
  Zur Kontrolle wird DE 0 auf H-Potential gelegt. Dabei müssen die Komparatorausgänge wieder auf L-Pegel gehen.

2.9.7. Einfluß der Umgebungstemperatur

Das thermische Verhalten des A/D-Wandlers wird im wesentlichen beeinflußt durch die Temperaturdrift

- der Offsetspannung der Operationsverstärker
- des Offsetstromes der Operationsverstärker
- der Spannungsteilerwiderstände der Operationsverstärker
- des D/A-Wandlers
- des Referenzelementes

Da der Temperaturkoeffizient nicht gerichtet ist, läßt sich der Einfluß der Schwankungen der Umgebungstemperatur im Arbeitstemperaturbereich auf die Genauigkeit des Analog-Digitalwandlers nur sehr schwer abschätzen.
Die thermische Drift der Offsetspannung und des Offsetstromes wirkt sich als zusätzliches Störsignal am Eingang der Verstärker aus. Dieses Störsignal kann berechnet werden, mit

$$\frac{\Delta U_e}{\Delta \vartheta} = \frac{\Delta U_{IO}}{\Delta \vartheta} + R \cdot \frac{\Delta I_{IO}}{\Delta \vartheta}$$

wobei R der im nicht negierenden Eingang liegende Kompensationswiderstand ist (z.B. R 702).
Die Gleichung zeigt, daß die Größe von R direkt eingeht.
Praktisch sollte angestrebt werden, diesen Wert möglichst klein zu wählen. Hierbei muß jedoch oft ein Kompromiß gewählt werden, da die im Spannungsteiler liegenden Widerstände den schaltungstechnischen Forderungen entsprechen müssen. Für den A 109 D und A 110 D werden keine Maximalwerte für die Temperaturdrift der Eingangsoffsetspannung und des Offsetstromes angegeben, so daß

eine Bestimmung dieses Störsignals nicht möglich ist. Wird eine minimale thermische Drift gefordert, sollten anstelle der Operationsverstärker A 109 und A 110 die Verstärker B 109 und B 110 eingesetzt werden.
Um den Einfluß der Temperaturdrift der Spannungsteilerwiderstände gering zu halten, sollten beispielsweise für R 701 und R 721 Widerstände mit möglichst kleinem TK verwendet werden. Wie bereits erwähnt, ist an dieser Stelle der Einsatz von Dünnschicht-Widerstandsnetzwerken des VEB KKWH zweckmäßig, da diese einen kleinen Differenz-Temperaturkoeffizienten aufweisen. Diese Dünnschicht-Widerstandsnetzwerke sind auch im D/A-Wandler-Netzwerk einzusetzen. Das Referenzelement der Referenzspannungsquelle sollte ebenfalls einen möglichst kleinen TK aufweisen. Für die verwendete SZY 23 wird ein TK $\leq 10 \cdot 10^{-6}$/grd angegeben. Im Bereich ϑ_a = 0...+ 70 °C errechnet sich damit theoretisch eine Spannungsänderung des Referenzelementes von

$$\Delta U = U_z/TK/ \cdot \Delta\vartheta$$
$$= 8{,}4 \text{ V} \cdot 10^{-5} \cdot 70$$
$$\Delta U = 6 \text{ mV}$$

Praktisch ist nach Angaben des Herstellers nur mit einer Spannungsänderung von ca. ± 1,5 mV, bezogen auf den Wert bei ϑ_a = 25 °C, zu rechnen. Damit der Temperatureinfluß auf den nachfolgenden Operationsverstärker der Referenzspannungsquelle gering bleibt, sind die bereits erwähnten Maßnahmen wie bei den übrigen Verstärkern durchzuführen.

2.9.8. Monotonie, Linearität

2.9.8.1. Monotonie

Die Monotonie des A/D-Wandlers wurde in folgender Form überprüft:

Gemäß der Tafel 2.1. werden an die Eingänge DE 0 bis DE 9 bestimmte Bitmuster gelegt und am Analogsignal-Ausgang AA mit einem Digitalvoltmeter die zugehörigen Spannungen gemessen. Die Bitmuster sind reine Binärwerte bzw. Werte, die um 1 bit niedriger liegen:

DE 9	DE 8	DE 7	DE 6	DE 5	DE 4	DE 3	DE 2	DE 1	DE 0	Binär-Wertig-keit	AA
H	H	H	H	H	H	H	H	H	L	1	4 mV
H	H	H	H	H	H	H	H	L	H	2	8 mV
H	H	H	H	H	H	H	H	L	L	3	12 mV
H	H	H	H	H	H	H	L	H	H	4	16 mV
H	H	H	H	H	H	H	L	L	L	7	28 mV
H	H	H	H	H	H	L	H	H	H	8	32 mV
H	H	H	H	H	H	L	L	L	L	15	60 mV
H	H	H	H	H	L	H	H	H	H	16	64 mV

usw.

Tafel 2.1.

Die Differenz der Ausgangsspannungen an AA muß 4 mV betragen.
Die Überprüfung der vorliegenden Schaltung ergab maximal eine Differenz von ± 2 mV vom Istwert.

2.9.8.2. Linearität

Die Überprüfung der Linearität erfolgte in folgender Form:

An die Digitaleingänge DE 0 bis DE 9 wurden bestimmte Bitmuster angelegt und an einen der Eingänge AE niederohmig eine Gleichspannungsquelle angekoppelt.
Die Eingangsspannung an diesem Eingang AE wird, von 0 V beginnend, so lange erhöht, bis sich am zugehörigen Ausgang DA ein L/H-Sprung ergibt. Die Eingänge SEL 1 und SEL 2 liegen dabei auf H-Potential. Das eingestellte Bitmuster, binär ausgedrückt und durch 100 dividiert, muß dann der an diesem Eingang AE angelegten Spannung entsprechen. Die Überprüfung wird zweckmäßig in gleichmäßigen Abständen über den gesamten Eingangsspannungsbereich von 10 V durchgeführt.
Beim vorliegenden Aufbau wurde festgestellt, daß die Linearität, gemessen über den gesamten Eingangsspannungsbereich, besser als 0,1 % ist (bezogen auf den Bereichsendwert).

2.9.9. Einschwingzeit

Die Einschwingzeit wurde in folgender Weise ermittelt:

- Beispielsweise an AE 1 wird ein Analogsignal von +5,12 V angelegt;
- Anlegen eines Binärsignals an die Eingänge DE 0...DE 9 abwechselnd mit einer Wertigkeit von 511 und 512 mit einer Frequenz von ca. 50 kHz;
- Mit Hilfe eines Zweistrahl-Oszilloskops wird die Zeitdifferenz zwischen der H/L-Flanke des Signals an DE 9 und der H/L-Flanke an DA 1 bzw. der L/H-Flanke DE 9 und L/H-Flanke an DA 1 bestimmt (SEL 1 und 2 auf H-Potential).

Beim vorliegenden Aufbau wurde eine Einschwingzeit von $< 10\ \mu s$ ermittelt.

2.9.10. Konstruktive Gesichtspunkte

Die hohe Empfindlichkeit des A/D-Wandlers erforderte die Berücksichtigung der folgenden wesentlichen Gesichtspunkte:

- großflächige Ausführung der Masseleitungen;
- Herausführen der Masseleitungen über mehrere Kontakte;
- Vermeidung von Masseschleifen;
- Abblocken der Betriebsspannungen auf der Leiterplatte mit 100 µF;
- Abblocken der Betriebsspannungsanschlüsse der Operationsverstärker mit 15 nF;
- Sorgfältige Siebung der positiven und insbesondere der negativen Betriebsspannung der Operationsverstärker.

2.9.11. Einsatz des A/D-Wandlers

Zum Betrieb einer Steckeinheit des A/D-Wandlers ist die Bereitstellung folgender Betriebsspannungen und -ströme erforderlich:

U_{CC} = 5 V ± 5 % (I_{CCtyp} ca. 70 mA, I_{CCmax} = 115 mA)

U_3 = +15 V ± 5 % (I_{3typ} ca. 130 mA, I_{3max} = 170 mA)

U_4 = -15 V ± 5 % (I_{4typ} ca. 190 mA, I_{4max} = 220 mA)

Die Anschlußbelegungen der Steckeinheit sind in Tafel A 2.15.7., die Anschlußbedingungen zum MRS 1 in Tafel A 2.16.8. aufgeführt.

2.10. Hinweise zur Baugruppenverdrahtung

Die zur Baugruppenverdrahtung notwendigen Verbindungen sind im Anhang in den Tafeln A 2.16. systematisch angegeben. Die in diesen Unterlagen verwendeten Anschlußbezeichnungen sind in Tafel A 2.13.1. definiert, Bild A 2.13.2. zeigt die Anschlußbelegung der Leiterkarten.
Bei der Erstellung der Verdrahtungsunterlagen wurde von einer Maximalkonfiguration des Mikrorechners ausgegangen, bei einer abgerüsteten Rechnervariante können alle nicht benötigten Verbindungen entfallen.
In Tafel A 2.16.8. sind alle Verbindungen angegeben, die beim Anschluß einer AD-Karte an den Rechner notwendig sind.
Hierbei ist durch den Anwender soft- und hardwaremäßig eine Spezifikation der OUT-Ports und deren Wertigkeit vorzunehmen.
Aus diesem Grunde ist auch nur der Anschluß einer Leiterkarte AD beschrieben; sollen mehr Karten eingesetzt werden, sind diese analog der 1. Karte zu verdrahten. Eine zahlenmäßige Beschränkung der AD-Karten auf 8 ergibt sich bei Parallelschaltung der Ausgänge DA dieser Leiterplatten.

2.11. Hinweise zur Inbetriebnahme des Rechners

2.11.1. Abgleichhinweise

Vor der ersten Inbetriebnahme des µR sind die Betriebsspannungen U_{CC}, U_{DD}, U_3, U_4 mittels der Einstellregler P 801...804 auf ihre Normwerte einzustellen. Eine Überprüfung dieser Spannungen an den entsprechenden Pins aller Steckverbinder ist zu empfehlen.
An der Funktionseinheit Anlauflogik der CPU-Baugruppe sind folgende Einstellungen vorzunehmen. Mittels Einstellregler P 201 wird die Ansprechschwelle der Unterspannungsabschaltung festgelegt. Hierzu ist an Meßpunkt M 201 eine Referenzspannung von 4,75 V einzustellen.
Mit dem Einstellregler P 202 kann das Tastverhältnis des Referenztaktsignals verändert werden. Dieses Verhältnis soll etwa 50 % betragen.

Abgleichhinweise zur Inbetriebnahme der AD-Baugruppe sind im Punkt 2.7. aufgeführt.

2.11.2. Prinzipielle Möglichkeiten zur Baugruppentestung

Nach erfolgter Überprüfung des Netzteils, ist als erstes die Funktion des Taktgenerators der CPU-Baugruppe zu testen (siehe auch Bild A 2.2.).
Bei der Baugruppenüberprüfung existiert die Möglichkeit der statischen Testung der Leiterkartenfunktionen außerhalb des Rechners. Eine weitere Möglichkeit ist eine entsprechende software-unterstützte Testung der Baugruppen, wenn die Prozessorbaugruppe und der erste Speicherblock der Speicherbaugruppe bereits arbeitsfähig sind. Hierzu muß vom Anwender ein Testprogramm erarbeitet werden, das beispielsweise innerhalb von ISRs bestimmte OUT-Belegungen ausgibt.
In diesem Zusammenhang und auch bei Programmtestungen ist der Anschluß einer SST-Baugruppe (siehe auch 2.3.2.3. und Bild A 2.1.) zu empfehlen. Hierbei ist die Anzeige der MAD'0...MAD'15 mittels LED's (über Treiberstufen D492 oder Transistoren) günstig.

3. Einführung in die Mikroprozessor-Programmierung

3.1. Allgemeines

3.1.1. Register

Der U 808 D enthält 7 frei verfügbare 8-bit-Speicherregister (Notizregister, Indexregister).
Die Register sind der Reihe nach bezeichnet mit A, B, C, D, E, H und L.
Register A ist der Akkumulator der CPU.
Die Register H und L haben zusätzlich noch die Funktion der Bereitstellung der Adresse für eine externe Speicheroperation (Lesen oder Schreiben). Dabei wird in Register H der obere Teil der Adresse (6 bit) und in Register L der untere Teil der Adresse (8 bit) zur Verfügung gestellt.
In der Mnemonie des U 808 D wird für den externen Speicherplatz die Bezeichnung M (Memory) verwendet. Softwaremäßig können die Register A bis L und der Speicherplatz M direkt angewählt werden.

3.1.2. Stack

Im U 808 D ist ein 8stufiger Adressen-Stapelspeicher (Stack) integriert. Jede Ebene des Stack ist 14 bit breit. Die oberste Stack-Ebene dient als Programmzähler (PC). Er wird während des Programmlaufes automatisch erhöht bzw. wird bei bestimmten Befehlen direkt oder von der unmittelbar darunter befindlichen Ebene beeinflußt. Die anderen sieben Stackebenen übernehmen bei bestimmten Befehlen den Inhalt der jeweils über- bzw. untergeordneten Ebene.
Im Stack können maximal 7 Rückkehradressen bei Anwendung von Unterprogrammen und Programmunterbrechungen abgespeichert werden.

3.1.3. Bedingungs-Flip-Flops

Die Bedingungs-Flip-Flops (engl.: Flags) C, Z, S und P werden bei bestimmten Operationen in Abhängigkeit vom Ergebnis der Operation gesetzt.
Hauptanwendungen finden die Bedingungs-Flip-Flops bei bedingten Entscheidungsoperationen.

Weitere Anwendungsfälle sind

für C: - binäre Addition und Subtraktion mit Übertrag
- Vergleich von Binärzahlen (< , >)
- Akkumulatorverschiebeoperationen

für Z: - Begrenzung bei Zähloperationen
- Vergleich von Binärzahlen (=)

für S: - Vorzeichenidentifikation für Zahlen im Wertebereich von -128 bis +127

für P: - Paritätsbitauswertung

3.1.4. Interrupt

Ein Interrupt ist eine Unterbrechung des laufenden Programms zu einem unvorhergesehenen Zeitpunkt.
Beim Eintreffen eines Interruptsignals wird das laufende Programm vorübergehend unterbrochen und eine Interrupt-Routine gestartet. Die Rückkehradresse kann im Stack abgelegt werden. Für den Wiedereintritt in das unterbrochene Programm wird dieser wieder in den Programmzähler übernommen.
Der U 808 D besitzt einen nichtmaskierbaren Interrupteingang. Da die Registerinhalte und die Bedingungs-Flip-Flop-Werte bei Interrupt nicht automatisch extern abgespeichert und am Ende rückgeladen werden, sind diese Operationen gegebenenfalls softwaremäßig in der Interrupt-Routine durchzuführen.

3.1.5. Speicher

Der U 808 D kann maximal 16 k byte Speicherkapazität direkt adressieren.
Innerhalb dieses Adressenbereiches können Programmspeicher (ROM) und Datenspeicher (RAM) beliebig angeordnet werden.
Die Adressierungsarten der Speicher sind die direkte Adressierung durch bestimmte Befehle über den Programmzähler und die indirekte Adressierung durch bestimmte Befehle über die Register H und L.

3.1.6. Eingabe/Ausgabe

Befehlsmäßig können 24 Ausgabetore und 8 Eingabetore zu jeweils 8 bit Breite direkt adressiert werden. Bei den direkt adressierenden Ein/Ausgabebefehlen wird immer mit dem Akkumulator operiert. Eine weitere Möglichkeit der Ein- bzw. Ausgabe von Daten ergibt sich in der Verwendung der indirekt adressierenden Memoryoperationsbefehle. In diesem Fall kann u.a. das höchste Bit der Speicheradresse zur Selektion zwischen Speicheradresse und Ein/Ausgabeadresse verwendet werden. Der maximal adressierbare Speicherbereich reduziert sich dabei auf 8 k byte, die anderen 8 k Adressen können beliebig für Eingabe- oder Ausgabeadressierung verwendet werden.
Eine Erhöhung der Anzahl der Eingabe-Tore wird dadurch möglich, daß vor einem IN-Befehl der Akkumulator mit einer zusätzlichen Adresse (Eingabe-Bank-Adresse) geladen wird. Diese Adresse wird bei Ausführung des IN-Befehls wärend T1 in das niederwertige Adressenregister des Mikrorechners geladen.
In T3 steht damit zusätzlich zur normalen IN/OUT-Adresse, die durch den Befehl adressiert wurde, eine weitere Adresse zur Verfügung, mit der eine Selektion einer 8-Tor-Gruppe aus mehreren solcher Eingabe-Tor-Gruppen (Bank) möglich wird (maximal 256 • 8).

3.2. Zahlensysteme

3.2.1. Allgemeines

Zahlen werden im allgemeinen in Zahlensystemen dargestellt. Verbreitet sind das Dualsystem, das Oktalsystem, das Dezimalsystem und das Hexadezimalsystem.
Allgemein gilt, daß in einem Zahlensystem zur Basis B die Ziffern 0 bis (B - 1) vorkommen.
Bei Systemen mit $B > 10$ müssen für die Ziffern 10, 11, 12... neue einstellige Symbole eingeführt werden (z.B. 10 = A, 11 = B usw.). In einem Zahlensystem zur Basis B können mit n Stellen B^n Zahlen dargestellt werden.

3.2.2. Dualsystem

Digitale Rechenanlagen arbeiten aus technischen Gründen im Dualsystem (B = 2). Die vorkommenden Ziffern sind 0 und 1. Bei einer mehrstelligen Dualzahl sind den einzelnen Positionen Zweierpotenzen zugeordnet.

Beispiel: $1001_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 9_{10}$

3.2.3. Zweierkomplement

Das Zweierkomplement ist für die Darstellung negativer Dualzahlen im U 808 D von Bedeutung. Mit einem 8-bit-Wort lassen sich $2^8 = 256$ Dualzahlen darstellen. Sollen positive und negative Zahlen erfaßt werden, so ist der Zahlenbereich aufzuspalten. Eine Darstellungsart negativer Dualzahlen ist das Zweierkomplement.

Das Zweierkomplement $\bar{Z}$ einer n-stelligen (positiven oder negativen) Dualzahl Z ist definiert als die Ergänzung von Z zur Zweipotenz 2^n

$$\bar{Z} = 2^n - Z$$

Ist Z positiv, so gibt $\bar{Z}$ die Darstellung von -Z an und umgekehrt liefert $\bar{Z}$ bei negativem Z den zugehörigen positiven Wert. Ein einfaches Verfahren zur Ermittlung des Zweierkomplementes besteht darin, die Dualzahl Z bitweise zu invertieren und anschließend 1 zu addieren.

Beispiel: Zweierkomplement von 28

	0001 1100	$= 28_{10}$
inv.	1110 0011	
+	1	
	1110 0100	$= -28_{10}$

Die Aufteilung des Zahlenbereiches ist beim Zweierkomplement nicht symmetrisch zu Null. Bei n = 8 geht der Zahlenbereich von -128 bis +127. Das höchstwertigste Bit einer 8stelligen Dualzahl kann als Vorzeichen aufgefaßt werden. 0 steht für positive und 1 für negative Zahlen.

3.2.4. Hexadezimalsystem

Die Basis des Hexadezimalsystems ist B = 16. Die Verwendung des Hexadezimalsystems gestattet es, die im U 808 D vorkommenden 8-bit-Befehls-Daten- und Adressenwörter durch zwei hexadezimale Ziffern darzustellen. Eine hexadezimale Ziffer gibt durch ein Zeichen den Zustand von vier Bits (Tetrade) an.
Die gültigen Ziffern des Hexadezimalsystems sind 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
Die Buchstaben A bis F entsprechen den Dezimalzahlen 10 - 15.

A = 10
B = 11
C = 12
D = 13
E = 14
F = 15

Die Umwandlung hexadezimal ⟶ dual ist dadurch gegeben, daß man jede Tetrade durch die vierstellige Dualzahl wiedergibt.

Beispiel: $E\,9_H = 1110\ 1001_2$

3.3. Befehle des U 808 D

3.3.1. Allgemeines

Alle Befehle und die für die Programmierung notwendigen Angaben dazu stehen in kompakter Form in den Befehlslisten im Anhang.
In diesem Abschnitt werden die Wirkung einzelner Befehle näher erläutert und auf Besonderheiten und Anwendungen eingegangen.

Die Abarbeitung eines Befehls läuft in folgenden Schritten ab:

Der Befehlszähler (Programmzähler) weist auf den auszuführenden Befehl. In der Befehlsholephase (PCI-Zyklus) wird das erste Byte, der Operationskode ausgelesen und interpretiert.
Im Operationskode ist die Operation und die Anweisung über die Befehlsform (1, 2 oder 3 byte) enthalten.
Anschließend werden, sofern es sich nicht um einen 1-byte-Befehl handelt, der Operand bzw. die Adressen-Bytes beschafft und der Befehl ausgeführt (PCR, PCC oder PCW-Zyklen). Gleichzeitig wird der Befehlszähler je nach Befehlslänge um 1, 2 oder 3 erhöht.

3.3.2. Adressierungsarten

Der U 808 D ermöglicht zwei Adressierungsarten. Bei der direkten Adressierung wird die Adresse des Speicherplatzes vom Befehl vorgegeben. Die direkte Adressierung hat nur Einfluß auf den Befehlsablauf (Programmsprünge bzw. Unterprogrammaufruf).
Bei der indirekten Adressierung wird die Adresse des Speicherplatzes, auf den zugegriffen werden soll, aus den Inhalten der Register H und L zusammengesetzt. Die indirekte Adressierung wird beim Lesen und Schreiben von Speicherplätzen verwendet (Memoryoperationen).

3.3.3. 8-bit-Transportbefehle

Die Transportbefehle, die man weiter in Lade- und Speicherbefehle unterteilen kann, dienen zum Bereitstellen der Operanden für die weitere Verarbeitung bzw. zum Abspeichern von Ergebnissen im Speicher.
Transportbefehle überschreiben den Inhalt des Bestimmungsregisters. Der Inhalt des Quellregisters bleibt unverändert.

3.3.4. Schiebebefehle

Es gibt zwei verschiedene Schiebebefehle (Bild 3.1)

- Kreisverschiebung Akku rechts/links
- Kreisverschiebung Akku rechts/links durch den Übertrag C

Man setzt Schiebefehle bei der seriellen bitweisen Verarbeitung eines Wortes ein.
Insbesondere finden sie Anwendung bei der binären Multiplikation bzw. Division.

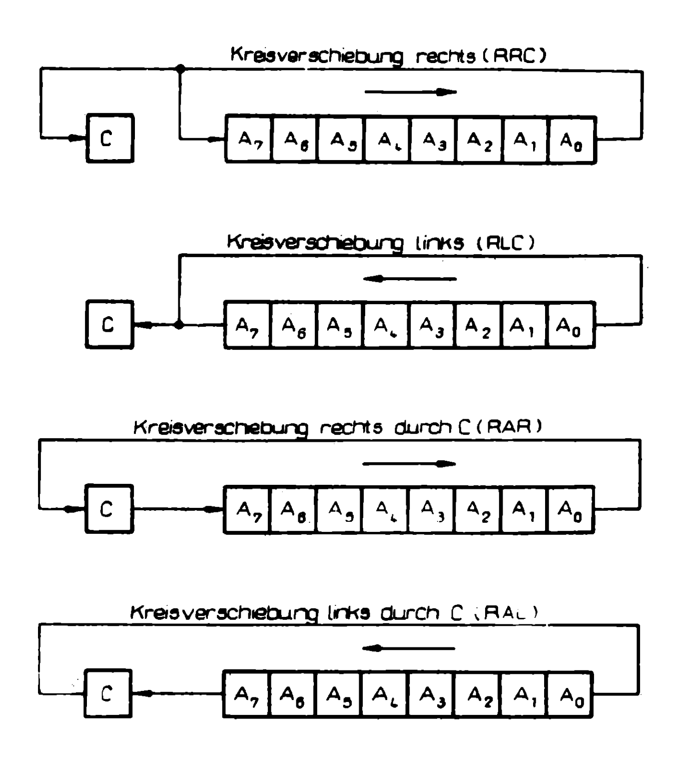


Bild 3.1: Wirkung der verschiedenen Schiebebefehle

3.3.5. Arithmetische Befehle

Es gibt die Befehle

- duale Addition/Subtraktion
- duale Addition/Subtraktion mit Berücksichtigung des Übertrages.

Die Additionsbefehle bzw. Subtraktionsbefehle verknüpfen zwei 8-bit-Dualzahlen nach den Regeln der dualen Addition bzw. Subtraktion. Die arithmetischen Befehle verwenden immer den Inhalt des Akkumulators als einen Operanden. Das Ergebnis der Operation steht ebenfalls wider im Akkumulator.

Beispiel: ADD (Interpretation als positive Dualzahlen)

1. Operand	0111 1100	124_{10}
2. Operand	+ 0001 0101	+ 21_{10}
Resultat	1001 0001	145_{10}

Beispiel: ADC (Interpretation als positive Dualzahlen)

1. Operand	0111 1001	121_{10}
2. Operand	+ 0010 1111	+ 47_{10}
Übertrag C (vorher)	1	+ 1
Resultat	1010 1001	169_{10}

Beispiel: SUB (Interpretation als Zweierkomplement)

1. Operand	0011 0010	50_{10}
2. Operand	- 0110 0100	- 100_{10}
Resultat	1100 1110	- 50_{10}

3.3.6. Logische Befehle

Es gibt Befehle für die drei logischen Operationen UND, ODER und exklusives ODER. Die logischen Befehle verwenden ebenfalls den Akkumulatorinhalt als ersten Operanden.

UND-Verknüpfung

Die UND-verknüpfenden Befehle verarbeiten die Operanden nach folgender Regel:

Op. 1	Op. 2	AND
0	0	0
0	1	0
1	0	0
1	1	1

Die Befehle werden verwendet, um gewisse Bitpositionen des Akkumulators unabhängig vom vorherigen Wert auf Null zu setzen, während andere Bits unverändert bleiben sollen. Der zweite Operand wird als Maske bezeichnet.

ODER-Verknüpfung

Die Befehle verknüpfen die Operanden nach folgender Regel:

Op. 1	Op. 2	OR
0	0	0
0	1	1
1	0	1
1	1	1

Mit den ODER-Befehlen können gewisse Bits des Akkumulators, durch die Maske gesteuert, unabhängig vom vorherigen Wert auf 1 gesetzt werden.

Exklusives ODER

Die bitweise Verknüpfung der beiden Operanden erfolgt gemäß:

Op. 1	Op. 2	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Mit Hilfe dieser Befehle und anschließender Negation kann man die logische Operation der Äquivalenz bilden. Die Negation des Akkumulators wird ausgeführt durch die Anwendung der exklusiv-ODER-Operation zwischen Akkumulator und dem Datenwort FF_H ($1111\ 1111_2$).
Der Akkumulatorinhalt kann gelöscht werden (Nullinhalt) durch Anwendung der exklusiv-ODER-Operation auf den Akkumulator selbst (XRA A).

3.3.7. Vergleichsbefehle

Die Vergleichsbefehle vergleichen den Inhalt des Akkumulators mit dem Vergleichsoperanden, indem intern die Differenz gebildet wird.
Die Vergleichsbefehle setzen entsprechend dem Vergleichsergebnis die Bedingungs-Flip-Flops, ohne die Operanden zu verändern.

3.3.8. Sprungbefehle

Die Sprungbefehle sind Grundlage für die Realisierung der wiederholten Ausführung von Programmteilen und das Abarbeiten verschiedener Programmzweige in Abhängigkeit von Entscheidungen.
Das automatische Fortschalten des Programmzählers nach jeder Befehlsausführung steuert den linearen Programmablauf. Sprungbefehle laden den Programmzähler mit einer neuen Adresse, wodurch im nächsten Befehlszyklus die Programmausführung ab der betreffenden Adresse fortgesetzt wird.

Unbedingter Sprungbefehl

Der unbedingte Sprungbefehl (JMP Adr.) lädt den Programmzähler mit der Sprungzieladresse. Die Programmausführung setzt in jedem Fall an dieser Stelle fort.

Bedingte Sprungbefehle

Der Befehlssatz umfaßt 8 bedingte Sprungbefehle. Es sind 4 Paare komplementärer Befehle. Mit ihnen kann jeweils ein Bedingungs-Flip-Flop abgefragt werden. Wenn die Bedingung erfüllt ist, wird der Sprung ausgeführt, d.h., in den Programmzähler wird die Adresse des Sprungziels geladen. Anderenfalls fährt das Programm mit dem nächsten Befehl fort.

3.3.9. Ruf- und Rückkehrbefehle

Rufbefehle sind Sprungbefehle, bei denen die Rückkehradresse für die Programmfortsetzung sichergestellt wird. Die Rückkehradresse ist die Adresse des Befehls, der dem Rufbefehl folgt.
Der U 808 D stellt zwei Rufbefehlstypen zur Verfügung, die sich in ihrer Adressierungsart unterscheiden:

- unbedingte und bedingte Rufbefehle
- Restartbefehle.

In jedem Fall wird die Rückkehradresse in die oberste Stackebene des Adressenstacks abgelegt, das Stackniveau um 1 erhöht und die Adresse des zu rufenden Unterprogramms in den Programmzähler geladen.

Unbedingter Rufbefehl

Der unbedingte Rufbefehl (CALL Adr.) bewirkt eine unbedingte Programmfortsetzung ab der Rufzieladresse.

Bedingter Rufbefehl

Der Befehlssatz umfaßt 8 bedingte Rufbefehle. Es sind 4 Paare komplementärer Befehle. Die Bedingungsverarbeitung entspricht der der bedingten Sprungbefehle.

Restartbefehle

Restartbefehle sind 1-byte-Rufbefehle. Der Befehlssatz der U 808 D enthält 8 Restartbefehle. Die Zieladressen der 8 Restartbefehle sind 8 absolute Adressen am Anfang des Adressenbereiches. Den Abschluß eines jeden Unterprogramms muß ein Rückkehrbefehl bilden. Rückkehrbefehle laden die in der obersten Stackebene abgelegte Adresse in den Programmzähler und erniedrigen das Stackniveau um 1.

Unbedingter Rückkehrbefehl

Der unbedingte Rückkehrbefehl (RET) bewirkt eine unbedingte Programmfortsetzung ab der Adresse der obersten Stackebene.

Bedingte Rückkehrbefehle

Der Befehlssatz umfaßt 8 bedingte Rückkehrbefehle. Es sind 4 Paare komplementärer Befehle. Die Bedingungsverarbeitung entspricht der der bedingten Sprungbefehle.

3.4. Programmiertechniken

3.4.1. Programmablaufplan (PAP)

Ein Programmablaufplan (Flußdiagramm) ist eine graphische Beschreibungsform eines Programms. Im Programmablaufplan sind insbesondere komplexe Zusammenhänge anschaulich darstellbar. Aufgrund ihres Übersichtscharakters empfiehlt es sich, bei der Programmentwicklung und zur Dokumentation von Programmen mit Programmablaufplänen zu arbeiten.
Für Programmablaufpläne werden folgende Sinnbilder verwendet (Bild 3.2).

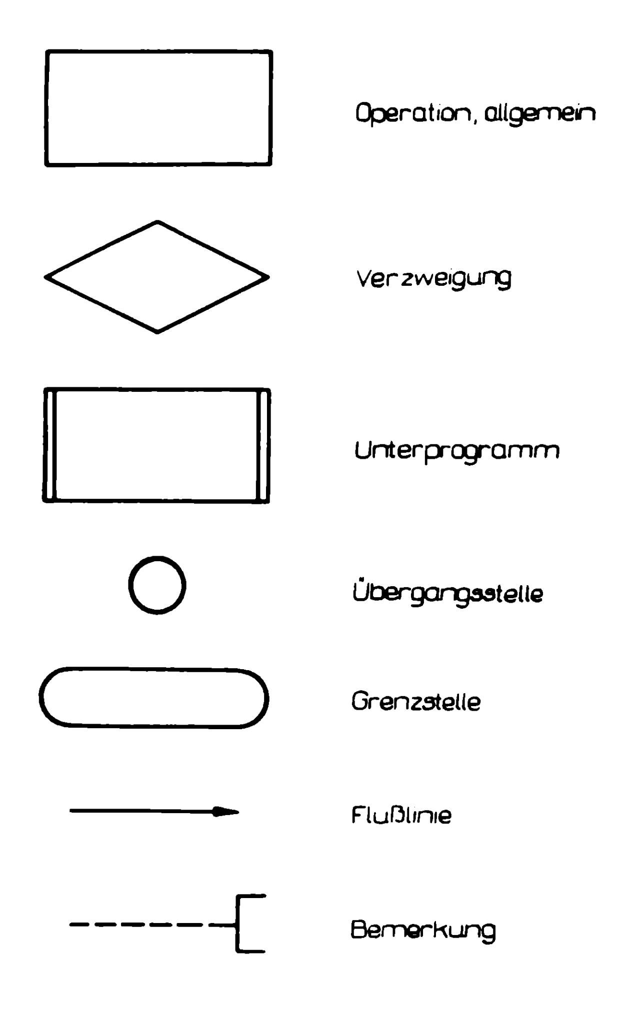


Bild 3.2: Sinnbilder für Programmablaufpläne

3.4.2. Bitmanipulationen

Unter Bitmanipulationen sind die Operationen

- Einblenden
- Ausblenden
- Zusammensetzen

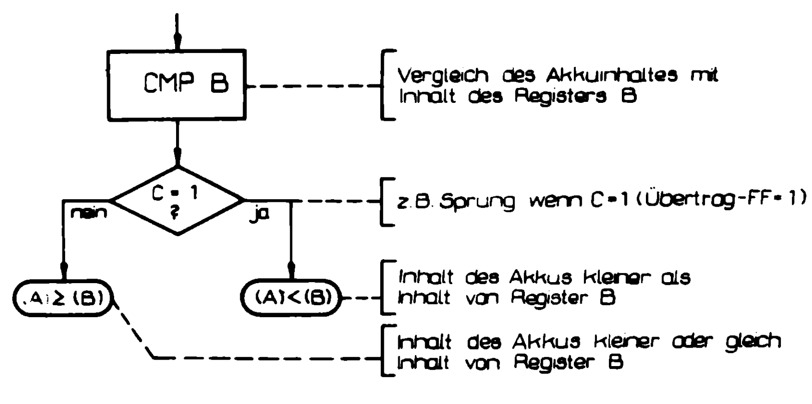
von einzelnen Binärstellen eines Wortes zu verstehen.
Als Befehle für solche Operationen eignen sich ODER-Verknüpfungen mit Maske (Einblenden), UND-Verknüpfungen mit Maske (Ausblenden) und UND- und ODER-Verknüpfungen mit Masken (Zusammensetzen).
Die Randbits eines Wortes sind günstigerweise auch durch Verschiebebefehle unter Berücksichtigung des Übertrages C kontrollierbar.

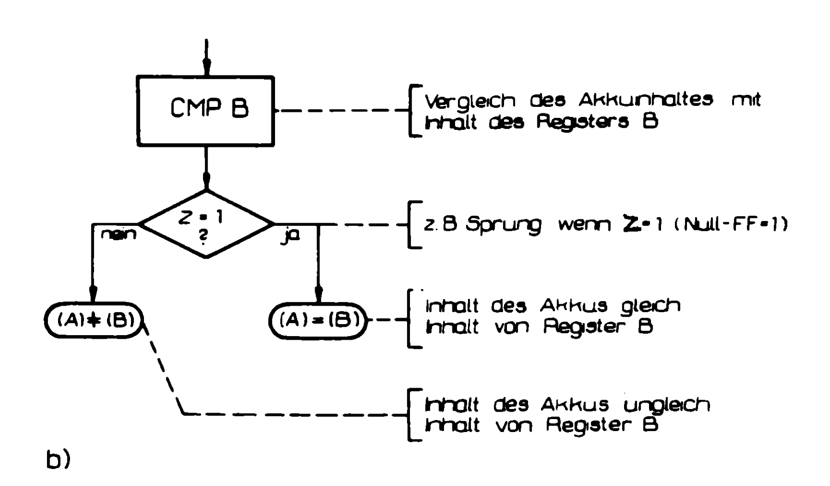
3.4.3. Programmverzweigungen

Bedingungen ermöglichen dem Programm, Entscheidungen auf Grund vorher errechneter Werte zu treffen. Solche Entscheidungen sind immer eindeutig mit ja oder nein zu beantworten. Die Abfrage der Bedingungen ergibt eine Programmverzweigung, deren Symbol zwei Ausgänge besitzt, die mit ja bzw. nein unterschieden werden.

Für den Vergleich von Zahlen kommen die Relationsoperationen $<$, $\leq$, $>$, $\geq$, $=$, $\neq$ (kleiner, kleiner oder gleich, größer, größer oder gleich gleich, ungleich) in Betracht. Als Befehle dafür werden die Vergleichsbefehle (CMP) in Verbindung mit bedingten Sprungbefehlen benutzt.

z.B. Vergleich des Akkumulatorinhaltes A mit dem Wort B (Bild 3.3).





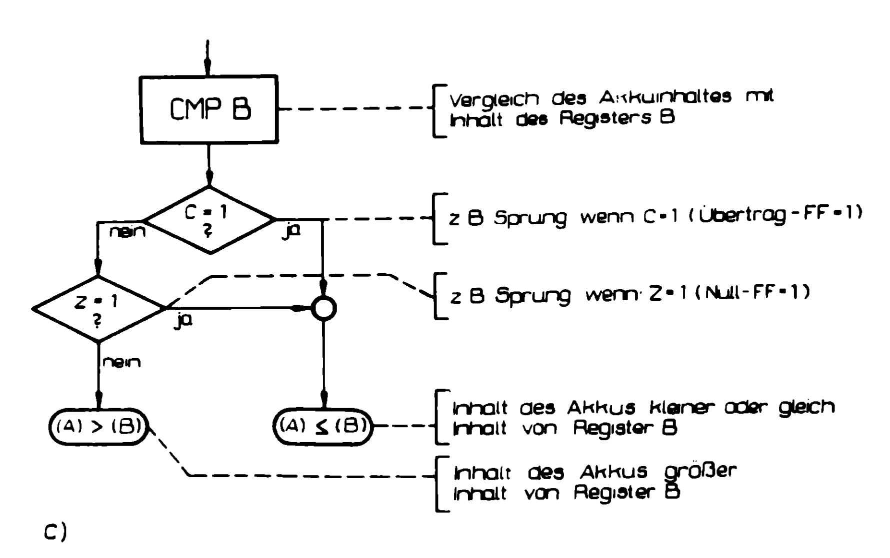


Bild 3.3: Prinzipielle Möglichkeiten des Zahlenvergleichs durch Programmverzweigungen

a) $(A) \geq (B)$, $(A) < (B)$

b) $(A) \neq (B)$, $(A) = (B)$

c) $(A) > (B)$, $(A) \leq (B)$

3.4.4. Programmschleifen

Schleifen dienen zur wiederholten Ausführung eines Programmstücks.
Schleifen werden unterschieden in Zählschleifen und Bedingungsschleifen. Bei Zählschleifen ist die Anzahl der Schleifendurchläufe vorher bekannt, Bedingungsschleifen werden abgebrochen, wenn die Bedingung erfüllt ist. Eine Schleife besteht im allgemeinen aus folgenden Komponenten:

- Initialisierung des Schleifenkriteriums
- Bearbeitung
- Veränderung des Schleifenkriteriums
- Endetest

Schleifen, die immer mindestens einen Schleifendurchlauf ergeben, lassen sich so aufbauen, daß der Endetest den Abschluß der Schleife bildet (Bild 3.4).

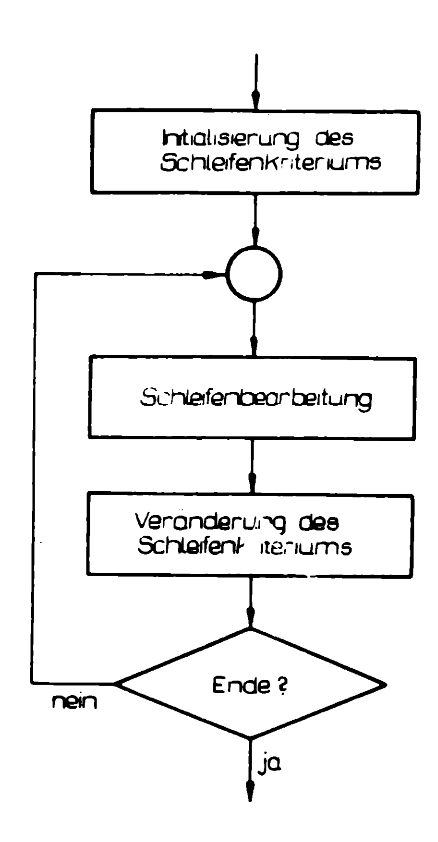


Bild 3.4: Schleifenaufbau für Schleifen mit mindestens einem Durchlauf

Falls die Zahl der Schleifendurchläufe auch Null sein kann, muß der Endetest bereits vor der ersten Bearbeitung ausgeführt werden (Bild 3.5).

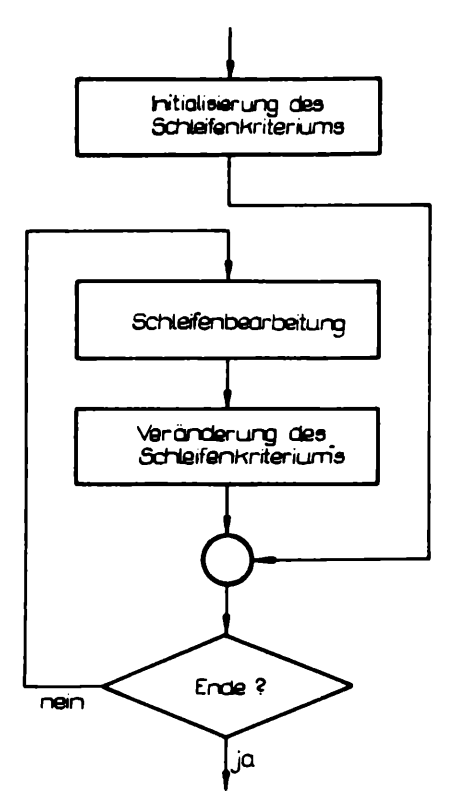


Bild 3.5: Schleifenaufbau für Schleifen mit nicht mindestens einem Durchlauf

3.4.5. Unterprogramme

Unterprogramme sind Teile des Gesamtprogramms und werden mit einem Rückkehrbefehl abgeschlossen. Ein Unterprogramm kann von einer beliebigen Stelle des Hauptprogramms aus aufgerufen werden. Bei Aufruf des Unterprogramms wird die Rückkehradresse gesichert. Unterprogramme werden eingesetzt, wenn

- die gleichen oder ähnliche Befehlsfolgen in einem Programm an mehreren Stellen vorkommen, und dadurch das wiederholte Einfügen dieses Programmstücks vermieden werden kann.
- es zweckmäßig ist, das Hauptprogramm übersichtlicher und kürzer zu gestalten, auch wenn manche abgeschlossene Programmstücke nur einmal vorkommen,
- auf Standardunterprogramme zurückgegriffen werden kann.

Die Aufrufmöglichkeiten von Unterprogrammen bleibt nicht auf das Hauptprogramm beschränkt. Ein Unterprogramm kann ein weiteres Unterprogramm aufrufen usw. (geschachtelte Unterprogramme).
Da der Stack des U 808 D maximal 7 Rückkehradressen aufnehmen kann, ist es möglich, bis zu 7 Unterprogrammen ineinander zu schachteln.

3.5. Interruptverarbeitung

Ein Interrupt stellt immer eine gewollte Unterbrechung des gerade in Arbeit befindlichen Programms dar.
Interrupts werden benutzt zum

- Einschieben von Interrupt-Service-Routinen (ISR's) in das laufende Programm,
- Starten und Stoppen von Programmabläufen.

Wird ein Programm für den Aufruf einer ISR unterbrochen, so muß in der Regel der gesamte Prozessorstatus (Bedingungs-Flip-Flops, Register A bis L) gerettet werden, um bei Rückeintritt in das unterbrochene Programm dieses richtig fortsetzen zu können.
Der U 808 D bietet keine direkte Möglichkeit, den Prozessorstatus zu retten.
Eine indirekte Methode besteht darin, die Registerinhalte nacheinander in den externen Speicher zu transportieren, durch logische Abfrage die Zustände der Bedingungs-Flip-Flops zu ermitteln und diese in Form eines Kodeworts ebenfalls im externen Speicher abzulegen.
Sollen alle Register gerettet werden, so sind zwei Registerinhalte in zwei Ausgabetoren zwischenzuspeichern, damit die Register H und L für die Adressierung des Speichers vorbereitet werden können. Die Ausgabetore sind ihrerseits mit zwei Eingabetoren zu verbinden, um die Rücknahme der Registerinhalte zu ermöglichen.
Die Wiederherstellung des Prozessorstatus am Ende der ISR geschieht in umgekehrter Reihenfolge. Sind die Bedingungs-Flip-Flops im Kodewort in der Form

$$CSS\overline{ZZP}OO$$

abgespeichert worden, so können die ursprünglichen Zustände der Flip-Flops durch Anwendung der Operation

ADD A

wiederhergestellt werden, nach dem das Kodewort in den Akkumulator geladen wurde.
Die Nachteile dieser Methode sind die relativ lange Abarbeitungszeit und die zwei reservierten Eingabe/Ausgabetore, die für andere Zwecke nicht mehr verwendet werden können.
Gibt es in einem System mehrere ISR's, die von außen durch Interrupts aufgerufen werden, so bieten sich für ihre Adressierung folgende Möglichkeiten:

1. Die indirekte Adressierung über Restartbefehle (max. 8 verschiedene ISR's)
2. Die direkte Adressierung mittels eingespeister unbedingter Rufbefehle (keine Begrenzung)
3. Die indirekte Adressierung über einen Restartbefehl, Sprung auf ein Unterprogramm, das die Startbedingung für die ISR erzeugt und Eingabe der Startadresse über ein oder zwei Eingabetore (keine Begrenzung).

In Bezug auf die Abspeicherung der Rückkehradresse im Stack sind Unterprogramme und ISR's gleichwertig. Deshalb ist darauf zu achten, daß bei gleichzeitiger Anwendung die maximale Stackbelegung von 7 Rückkehradressen nicht überschritten wird.

3.6. Grundsätzliche Hinweise zur Mikroprozessor-Programmierung

Mikroprozessor-Programmierung heißt:

Das Lösen eines Problems durch das Finden eines Lösungsverfahrens und die Realisierung dieses Verfahrens in Form eines Programms für einen verfügbaren Mikroprozessor. Die einzelnen Phasen der Programmierung sind:

- Aufgabenspezifikation
- Suchen des Lösungsverfahrens
- Programmkonstruktion
- Test
- Programmdokumentation

In der Phase der Aufgabenspezifikation wird aus der vorliegenden Aufgabenstellung eine exakte, schriftlich fixierte Beschreibung der gewünschten Leistung eines Programms unter Einbeziehung aller Rahmenbedingungen erarbeitet.
Wichtige Teilergebnisse dieser Arbeitsphase sind:

- Festlegung über Umfang und Allgemeingültigkeit des anzustrebenden Verfahrens
- Abschätzungen über Verallgemeinerungs- bzw. Universalitätsmöglichkeiten des Programms
- Einschätzung des Speicherbedarfes und des E/A-Tor-Bedarfes
- Analyse kritischer Zeitstellen.

Unmittelbar an diese Etappe schließt sich die Phase der Suche nach einem geeigneten Lösungsverfahren an.
Je nach Aufgabenstellung kann man auf bekannte Algorithmen zurückgreifen (meist mathematische Aufgaben) oder es müssen Lösungsalgorithmen erarbeitet werden (meist organisatorische Aufgaben). Wichtige Hilfsmittel hierbei sind Flußdiagramme, die besonders bei komplexen Abläufen eine übersichtliche Darstellung des Programm- bzw. Datenflusses ermöglichen. Oft ist es günstig das Problem in Teilprobleme zu zerlegen und diese nacheinander zu lösen, dabei gilt für Teilprobleme der gleiche Arbeitsablauf wie für das Gesamtproblem.
Teilergebnisse des Lösungsverfahrens sind:

- Fixierung der Lösungsalgorithmen
- Darstellung mittels Flußdiagramme
- Zerlegung in nacheinander zu lösende Teilprobleme.

Die Ergebnisse der Suche des Lösungsverfahrens sind Voraussetzung für die Programmkonstruktion.
In der Phase der Programmkonstruktion wird die Programmarchitektur geplant, das Programm kodiert und mittels technischer Programmierhilfen (Entwicklungssystem, Wirtsrechner) und der entsprechenden Softwareunterstützung (Assembler, Compiler) in Maschinenprogramme übersetzt.

Bei der Programmkonstruktion sind folgende Grundsätze zu beachten:

- Programme sollten leicht zu bedienen und übersichtlich konstruiert sein.
- Große Programme sind in funktionelle Teilprogramme zu zerlegen.
- Die Leistungen und Schnittstellen der Teilprogramme sind gut zu dokumentieren.
- Schnittstellen sind nach Möglichkeit zu standardisieren, wodurch sich leichte Anschließbarkeit und Austauschbarkeit von Programmteilen ergibt.
- Eingaben in das Programm sollten angefordert und quittiert werden, um Fehlbedienung weitestgehend auszuschließen.
- Das Programm ist gegen fehlerhafte Handhabung abzusichern. Eingegebene Daten sind zu prüfen.
- Übersichtlichen und dadurch meist längeren Programmen ist der Vorrang gegenüber eleganteren aber schwer zu verstehenden Programmen zu geben, wenn keine anderen Einschränkungen dies verbieten.
- Beim Programmentwurf sind geeignete Testmöglichkeiten vorzusehen, wodurch der Zeitaufwand in der Testphase reduziert werden kann.

Der Programmtest fördert die vorhandenen Programmfehler zu Tage und ermöglicht nach deren Beseitigung den Nachweis des fehlerfreien Arbeitens des Programms.
Als Testhilfen können dafür wieder die technischen Programmierhilfen (Entwicklungssystem, Wirtsrechner) in Verbindung mit der entsprechenden Testsoftware (Simulatorprogramm, Debuggprogramm) eingesetzt werden.
Da Programme beim ersten Testlauf im allgemeinen nicht fehlerfrei sind, ist ein systematisches Vorgehen bei der Fehlersuche sehr wichtig.
Es ist sinnvoll das Programm erst per Hand zu kontrollieren und dann mit einer Reihe ausgewählter Testdaten zu starten. Die Testdaten sind so zu variieren, daß jeder Programmzweig mindestens einmal durchlaufen wird. Insbesondere ist auf die richtige Behandlung von Grenzwerten und Fehlersituationen zu achten.

Ein Testrahmen für die Ausgabe zahlreicher Zwischenergebnisse erleichtert und verkürzt die Fehlersuche.
Schwerwiegende Fehler im logischen Konzept führen zurück zur Phase der Programmkonstruktion, wobei dieser Zyklus durchaus mehrmals durchlaufen werden kann.
Den Abschluß der Testarbeit bildet im allgemeinen ein Test des Programms in einem Prototyp-Mikrorechner. In diesem Test kann die Richtigkeit des Programms in Bezug auf die Lösung der Aufgabe nachgewiesen werden.
Die Programmdokumentation ist ein wichtiger Bestandteil der Programmierarbeit und sollte gleichzeitig mit dem Programm entstehen.
Die Grundlage der Dokumentation bildet eine übersichtlich gegliederte, durch sachbezogene Namensgebung und Kommentare gut lesbar gestaltete Programmliste.
Weiterhin gehören dazu:

- Allgemeine Programmbeschreibung, Leistung des Programms
- Bedienungsanleitung
- Verfahrensbeschreibung
- Teilprogrammbeschreibung
- Programmablaufpläne
- Speicher- und Torbelegungspläne
- Hinweise auf mögliche Erweiterungen
- Hinweise auf mögliche Verbesserungen bzw. Schwachstellen.

4. Beispiele

4.1. 10-Bit-A/D-Wandlung nach dem Prinzip der sukzessiven Approximation

Für den im Abschnitt 2.9. gezeigten A/D-Wandler wird hier ein Steuerprogramm vorgestellt, das nach dem Prinzip der sukzessiven Approximation arbeitet. Der A/D-Wandler enthält maximal acht Komparatoren. Die Selektion der einzelnen Komparatorausgänge erfolgt durch das Programm mittels einer Maske.

Die Ansteuerung des D/A-Umsetzers geschieht über zwei Ausgabe-Tore (10 bit), die acht Komparatoren werden über ein Eingabe-Tor (8 bit) abgefragt (Bild 4.1).

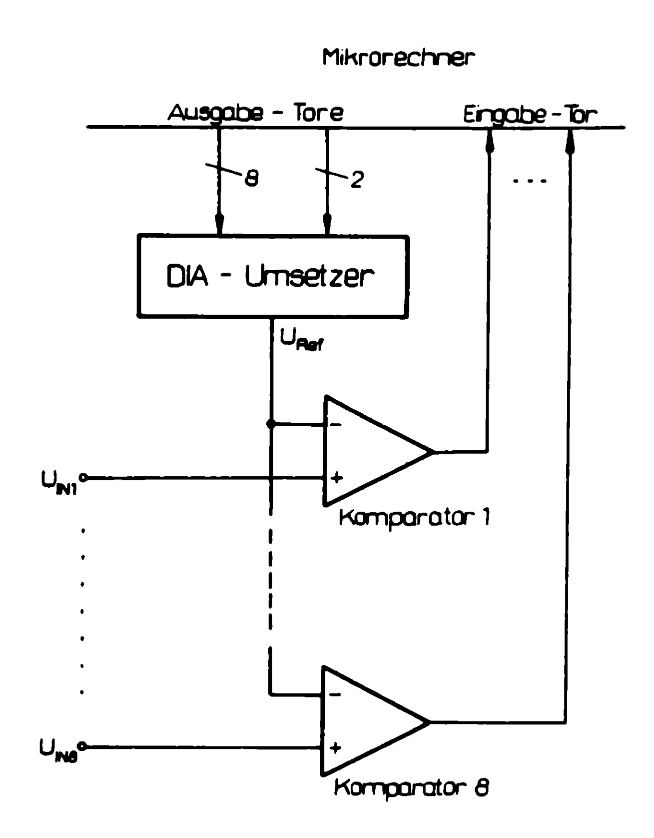


Bild 4.1.: Prinzipaufbau des A/D-Wandlers

Das Steuerprogramm arbeitet wie folgt (Bild 4.2.):

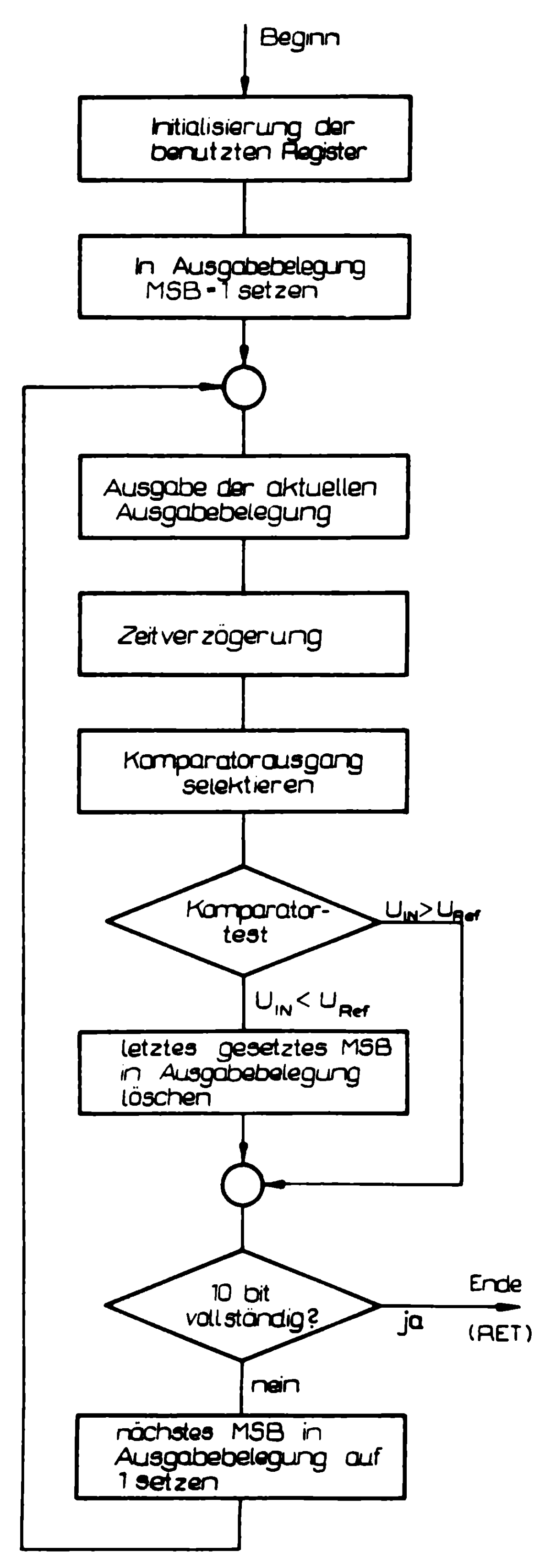


Bild 4.2.: Flußdiagramm für den prinzipiellen A/D-Wandlungs-Algorithmus

Zuerst schaltet der Rechner das MSB (engl.: most significant bit) in der Ausgabebelegung ein, wartet bis der D/A-Umsetzer und die Koparatoren eingeschwungen sind, liest dann den Zustand der Komparatoren ein und selektiert einen Komparator durch die Maske. Zeigt der Komparator, daß die MSB-Spannung kleiner als der entsprechende Analogeingang ist, dann bleibt das MSB im Ein-Zustand und das nächstkleinere MSB wird zusätzlich eingeschaltet. Zeigt jetzt der Komparator, daß der Analogwert der beiden MSB's größer ist als das Eingangsanalogsignal, wird das 2. MSB wieder ausgeschaltet und dafür das 3. MSB eingeschaltet. Auf diese Weise werden alle 10 Bit des A/D-Wandlers durchgetestet. Am Ende der Umwandlung entspricht der Eingang des D/A-Umsetzers auf $\pm$ 1 LSB genau dem Eingangsanalogsignal. Dieser Wert wird in zwei Registern (B und C) des Prozessors gespeichert und kann weiter verarbeitet werden.
Die totale Umsetzzeit für 10 bit beträgt minimal 3,0 ms und maximal 3,6 ms.
Das für die Wandlung erforderliche Steuerprogramm ist als Unterprogramm ausgeführt und benötigt insgesamt 48 Programmspeicherplätze.
In Tafel 4.1 ist die vollständige Befehlsliste für das Unterprogramm angegeben.

Tafel 4.1: Befehlsliste für 10-bit-A/D-Wandlung

```
; ADU: 10-Bit-A/D-Wandlung (sukzessive Approximation)
; Ergebnisspeicher B: niederwertiges Byte (8 bit)
; "                C: höherwertiges Byte  (2 bit)
; Verschiebeoperator: D
; Komparatormasken KOMA (01, 02...80) : E
; Ausgabe-Tor niederw. Ausgabebelegung (8 bit) : AT 1
; Ausgabe-Tor höherw.  Ausgabebelegung (2 bit) : AT 2
; Eingabe-Tor für 8 Komparatoren: ET 1
; Stackebenen : 0

      ; Beginn des Programms
      ; Initialisierung der Register (KOMA im übergeordneten
      ; Programm gesetzt)
ADU: XRA A           ; - Löschen Akku
     MOV C, A        ; - Löschen C
     MVI D, 02 H     ; - Verschiebeoperator setzen
     OUT AT 1        ; - Ausgabe niederw. Byte
     ; höherwertige Schleife (2 bit)
ADU 1: MOV A, C      ; - Vorbereiten höherw. Byte
       ADD D         ; für Komparatortest
       OUT AT 2      ; - Ausgabe höherw. Byte
       NOP           ; - Wartebefehl (20 µs) = MOV A, A
       IN ET 1       ; - Komparatorabfrage
       ANA E         ; - Ausblenden eines Komparators
       JZ ADU 2      ; - Sprung wenn Null
       MOV A, C      ; - Setzen des getesteten MSB
       ADD D         ; im höherw. Byte
       MOV C, A
ADU 2: MOV A, D
       RAR           ; - Verschiebeoperator nach rechts schieben
       MOV D, A
       JNC ADU 1     ; - Sprung wenn nooh nicht 2 bit getestet
       MOV A, C
       OUT AT 2      ; - Ausgabe höherw. Byte
       ; niederwertige Schleife (8 bit)
```

```
         MVI D, 80 H  ; - Verschiebeoperator setzen
         XRA A        · - Löschen Akku
         MOV B, A     ; - Loschen B

ADU 3:   MOV A, B     ; - Vorbereiten niederw. Byte
         ADD D        ; für Komparatortest
         OUT AT 1     ; - Ausgabe n ederw. Byte
         NOP          : - Wartebefehl (20 µs) = MOV A, A
         IN ET 1      ; - Komparatorabfrage
         ANA E        ; - Ausblenden eines Komparators
         JZ ADU 4     ; - Sprung wenn Null
         MOV A, B     ; - Setzen des getesteten MSB
         ADD D        ; im niederw. Byte
         MOV B, A

ADU 4:   MOV A, D     ; - Verschiebeoperator nach
         RAR          ; rechts schieben
         MOV D, A
         JNC ADU 3    ; - Sprung wenn noch nicht 8 bit getestet
         RET          ; - Ende des Programmes
```

4.2. Binär/BCD-Umwandlung nach dem Prinzip der bedingten Dreierkorrektur

Eine Standardtechnik für die Binär/BCD-Umwandlung ist der 3er-Additions-Algorithmus. Die Bilder 4.3 und 4.4 zeigen das Flußdiagramm und das Prinzip der Verschiebung und der bedingten Dreierkorrektur.

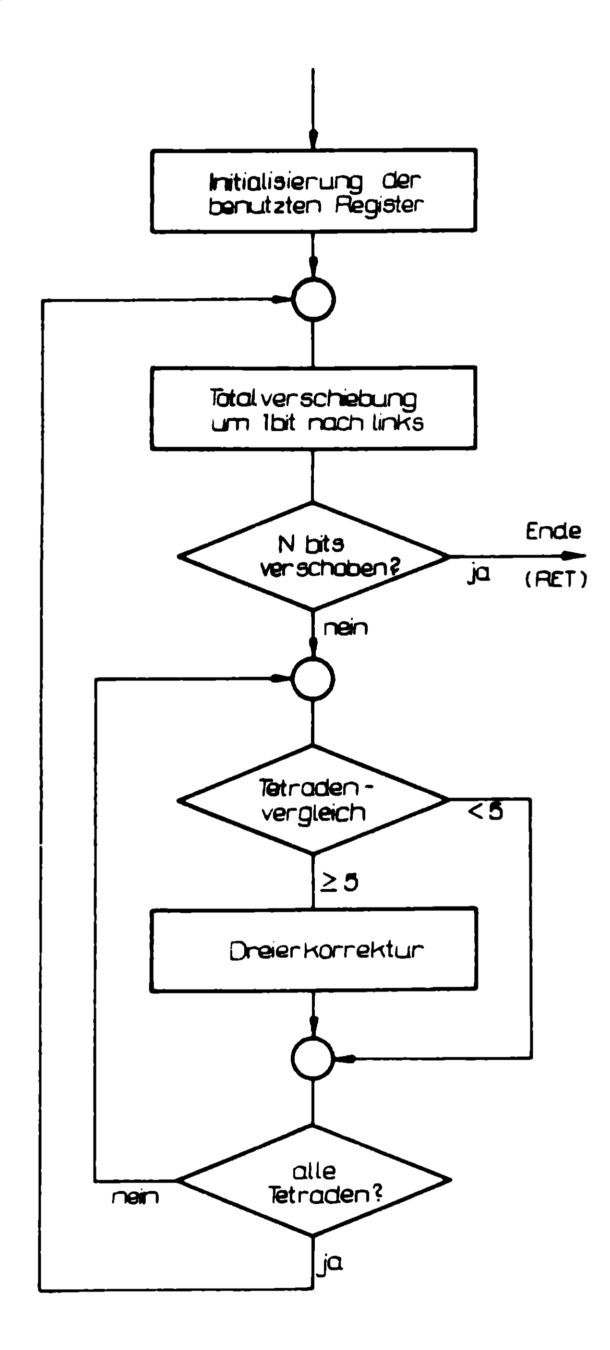


Bild 4.3: Flußdiagramm für die Binär/BCD-Umwandlung nach dem Prinzip der bedingten Dreierkorrektur

	Hunderter		Zehner				Einer				8-bit-Binärzahl						
											1	1	1	1	1	1	1
Rechtsverschiebung										1	1	1	1	1	1	1	1
Rechtsverschiebung									1	1	1	1	1	1	1	1	
Rechtsverschiebung								1	1	1	1	1	1	1	1		
Add. 3 zu Einern							1	0	1	0	1	1	1	1	1		
Rechtsverschiebung						1	0	1	0	1	1	1	1	1			
Add. 3 zu Einern						1	1	0	0	0	1	1	1	1			
Rechtsverschiebung					1	1	0	0	0	1	1	1	1				
Rechtsverschiebung				1	1	0	0	0	1	1	1	1					
Add. 3 zu Zehnern			1	0	0	1	0	0	1	1	1	1					
Rechtsverschiebung		1	0	0	1	0	0	1	1	1	1						
Add. 3 zu Einern		1	0	0	1	0	1	0	1	0	1						
Rechtsverschiebung	1	0	0	1	0	1	0	1	0	1							
Anzahl der Verschiebungen 8	2		5				5										

Bild 4.4: Prinzip der Binär/BCD-Umwandlung (Beispiel $FF_H \rightarrow 255$)

Der Ausgangspunkt der Umwandlung ist ein N-bit-Binärwort. Das Binärwort wird N-mal nach links in die Ergebnisregister verschoben. Die Ergebnisregister sind eingeteilt in mehrere Tetraden in denen am Ende der Umwandlung die BCD-Zahlen in Form gepackter Operanden vorliegen.
Nach jeder Verschiebung um ein bit wird bis zur (N-1)-ten Verschiebung jede Tetrade der Ergebnisregister daraufhin kontrolliert, ob ihr Inhalt größer oder gleich 5 ist. Sollte das der Fall sein, wird zu der entsprechenden Tetrade 3 addiert.
Nach der N-ten Verschiebung liegt das Ergebnis der Umwandlung vor. Die höchstwertigste BCD-Zahl ist am weitesten links und die niederwertigste ganz rechts in den Ergebnisregistern lokalisiert.
Das für die Umwandlung von maximal 4 Dekaden (N = 16) erforderliche Programm ist als Unterprogramm ausgeführt und benötigt insgesamt 65 Programmspeicherplätze.
Tafel 4.2. zeigt die Befehlsliste für das Programm.

Tafel 4.2: Befehlsliste für Binär/BCD-Umwandlung

```
; Binär/BCD-Umwandlung (16 bit)

; Binärzahl niederwertiges Byte: B
; Binärzahl höherwertiges Byte : C
; Ergebnisspeicher D, E
; Ergebnis Einer               : D (2^0...2^3)
; "        Zehner              : D (2^4...2^7)
; "        Hunderter           : E (2^0...2^3)
; "        Tausender           : E (2^4...2^7)
; Zyklusszähler                : H
; Zwischenspeicher             : L
; Stackebenen                  : 1

            ; Beginn des Programms
           ; Initialisierung der Register
BCDKO : XRA A               ; - Löschen A
        MOV D, A            ; - Löschen Ergebnisspeicher D
        MOV E, A            ; - Löschen Ergebnisspeicher E
        MVI H, 10 H         ; - Setzen Zyklusszähler

           ; Schleife Totalverschiebung

ROTAT : MOV A, B
        RAL                 ; - B um 1 bit nach links schieben
        MOV B, A
        MOV A, C
        RAL                 ; - C um 1 bit nach links schieben
        MOV C, A
        MOV A, D
        RAL                 ; - D um 1 bit nach links schieben
        MOV D, A
        MOV A, E
        RAL                 ; - E um 1 bit nach links schieben
        MOV E, A

           ; Zyklussteuerung

        DCR H               ; - Zyklusszähler erniedrigen
        RZ                  ; - Ende des Programms wenn
                              16 Zyklen
```

```
           ; bedingte Dreierkorrektur

        MOV A, D                ; - Aufruf der bedingten Dreier-
        CALL KORR               ;   korrektur für D
        MOV L, A
        MOV D, E                ; - Aufruf der bedingten Dreier-
        MOV E, L                ;   korrektur für E
        MOV A, D
        CALL KORR
        MOV D, E
        MOV E, A
        JMP ROTAT               ; - Sprung auf Programmanfang

          ;Korrektur
KORR:   ANI OFH                 ; - Abtrennen untere Tetrade
        CPI 05H                 ; - Test: Tetrade ≥ 5 ?
        JC KORR 1               ; - Sprung wenn   < 5
        ADI 03H                 ; - Addiere 3 wenn ≥ 5

KORR 1: MOV L, A                ; - Abtrennen obere Tetrade
        MOV A, D
        RRC
        RRC
        RRC
        RRC
        ANI OFH
        CPI 05H                 ; - Test: Tetrade ≥ 5 ?
        JC KORR 2               ; - Sprung wenn   < 5
        ADI 03H                 ; - Addiere 3 wenn ≥ 5

KORR 2: RLC                     ; - Zusammenfügen der korrigierten
        RLC                     ;   Tetraden
        RLC
        RLC
        ORA L
        RET                     ; - Ende Unterprogramm Korrektur
```

4.3. Binäre Multiplikation

Als sehr geeignetes Verfahren für die binäre Multiplikation erweist sich das Prinzip der Summation von Partialprodukten, die mittels Verschiebung erzeugt werden. Das Verfahren hat gegenüber anderen den Vorteil, daß es verhältnismäßig wenig Operationszeit benötigt und daß die Zeit annähernd unabhängig von der Größe der Operanden ist.
Die Multiplikation eines binären Wertes mit Potenzen von 2 (2^n) läßt sich sehr einfach durch Linksverschiebung des Wertes erreichen.
Die Anzahl der notwendigen Verschiebungen um jeweils ein bit entspricht dabei der Potenz n.

Beispiel: Binäre Multiplikation $57 \cdot 2^2 = 228$

57	$\cdot$	2^2
1. Operand (Akkumulator)	$\cdot$	2. Operand (beliebiges Register)
00111001	$\cdot$	00000100
01110010	<--	1. Verschiebung links
11100100	<--	2. Verschiebung links
228		

Entsprechend der Zuordnung von Zweierpotenzen für die einzelnen Bitpositionen von Dualzahlen läßt sich eine allgemeine binäre Multiplikation auf eine Summation von Partialprodukten von Potenzen von 2 zurückführen:

$$(A_7 \cdot 2^7 + \ldots + A_0 \cdot 2^0) \cdot (B_7 \cdot 2^7 + \ldots + B_0 \cdot 2^0) =$$
$$(A_7 \cdot 2^7 + \ldots + A_0 \cdot 2^0) \cdot B_7 \cdot 2^7 + (A_7 \cdot 2^7 \ldots + A_0 \cdot 2^0)$$
$$\cdot B_6 \cdot 2^6 + \ldots + (A_7 \cdot 2^7 + \ldots + A_0 \cdot 2^0) \cdot B_0 \cdot 2^0$$

Den prinzipiellen Funktionsablauf für eine allgemeine binäre Multiplikation zweier 8-bit-Faktoren zeigt Bild 4.5.

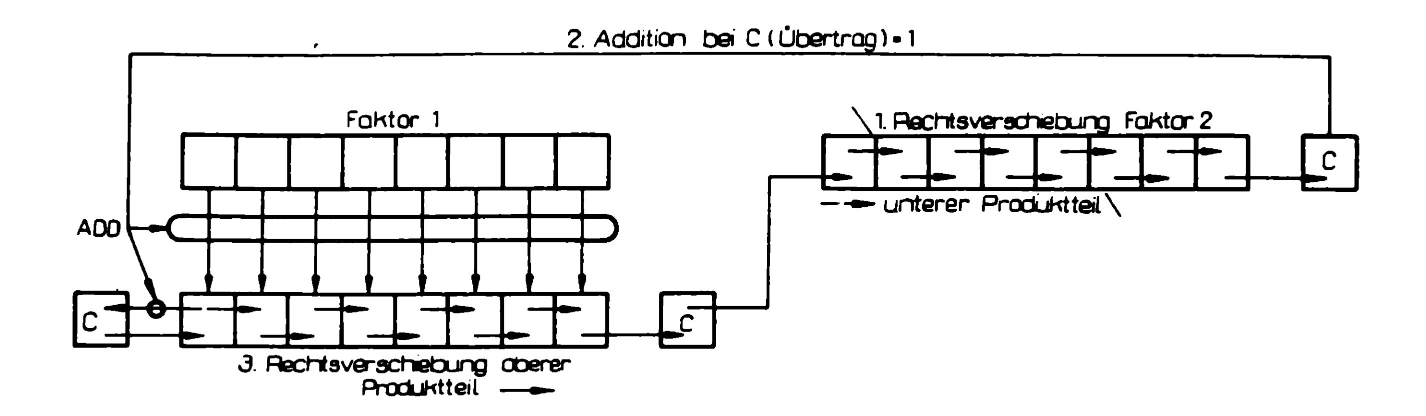


Bild 4.5: Prinzipieller Funktionsablauf für die allgemeine binäre Multiplikation von zwei 8-bit-Faktoren mittels Verschiebung und Partialproduktsummation

Die Wertigkeitsverschiebung eines Faktors wird dadurch realisiert, daß die Ergebnisregister relativ zu dem Faktor 1 nach rechts verschoben werden und die Summation der Partialprodukte, gesteuert durch den zweiten Faktor, immer mit dem höheren 8-bit-Produktteil durchgeführt wird.

Tafel 4.3 zeigt die Befehlsliste für das Multiplikationsprogramm, das als Unterprogramm aufgebaut ist und 17 Programmspeicherplätze benötigt.

Tafel 4.3: Befehlsliste für die binäre Multiplikation von zwei 8-bit-Faktoren

```
; BIMU: Binäre Multiplikation (2 x 8 bit)
;        Faktor 1 : B
;        Faktor 2 : D
;        oberer Produktteil : C
;        unterer Produktteil : B
;        Zyklussteuerung     : E
;        Stackebenen         : 0
                ; Beginn des Programms
                ; Initialisieren der benötigten Register
                ; Faktoren durch das übergeordnete Programm gesetzt
BIMU   :        MVI E, 09H      ; - setzen Zykluszähler

                ; Schleife

BIMU 1 :        MOV A, B        ; - Rechtsschieben Faktor 1
                RAR
                MOV B, A
                DCR E           ; - Zykluszähler erniedrigen
                RZ              ; - Programmende nach 9 Zyklen
                MOV A, C
                JNC BIMU 2      ; - Sprung wenn von Faktor 1 kein
                                ;   Übertrag
                ADD D           ; - Addiere D zu C wenn Übertrag

BIMU 2 :        RAR             ; - Rechtsschieben Faktor C
                MOV C, A
                JMP BIMU 1      ; - Sprung auf Schleifenanfang
```

4.4. Zeitverzögerungen

Der U 808 D wird mit einem quarzstabilisierten Taktgenerator betrieben, dessen Grundfrequenz 1/3 MHz bis 1/2 MHz betragen kann. Wird beispielsweise die Taktfrequenz zu 0,5 MHz festgelegt, so benötigt der Prozessor für die Abarbeitung jedes maschineninternen Zustandes 4 µs. Da für jeden Befehl die Anzahl der benötigten Zustände bekannt sind, lassen sich durch Abarbeitung geeigneter Befehlsfolgen Zeitverzögerungen in Länge von n x 4 µs erzeugen. Als geeignete Befehlsfolgen für Zeitverzögerungen erweisen sich Zählschleifen.

Bild 4.6 zeigt eine einfache und eine doppelte Verzögerungsschleife.

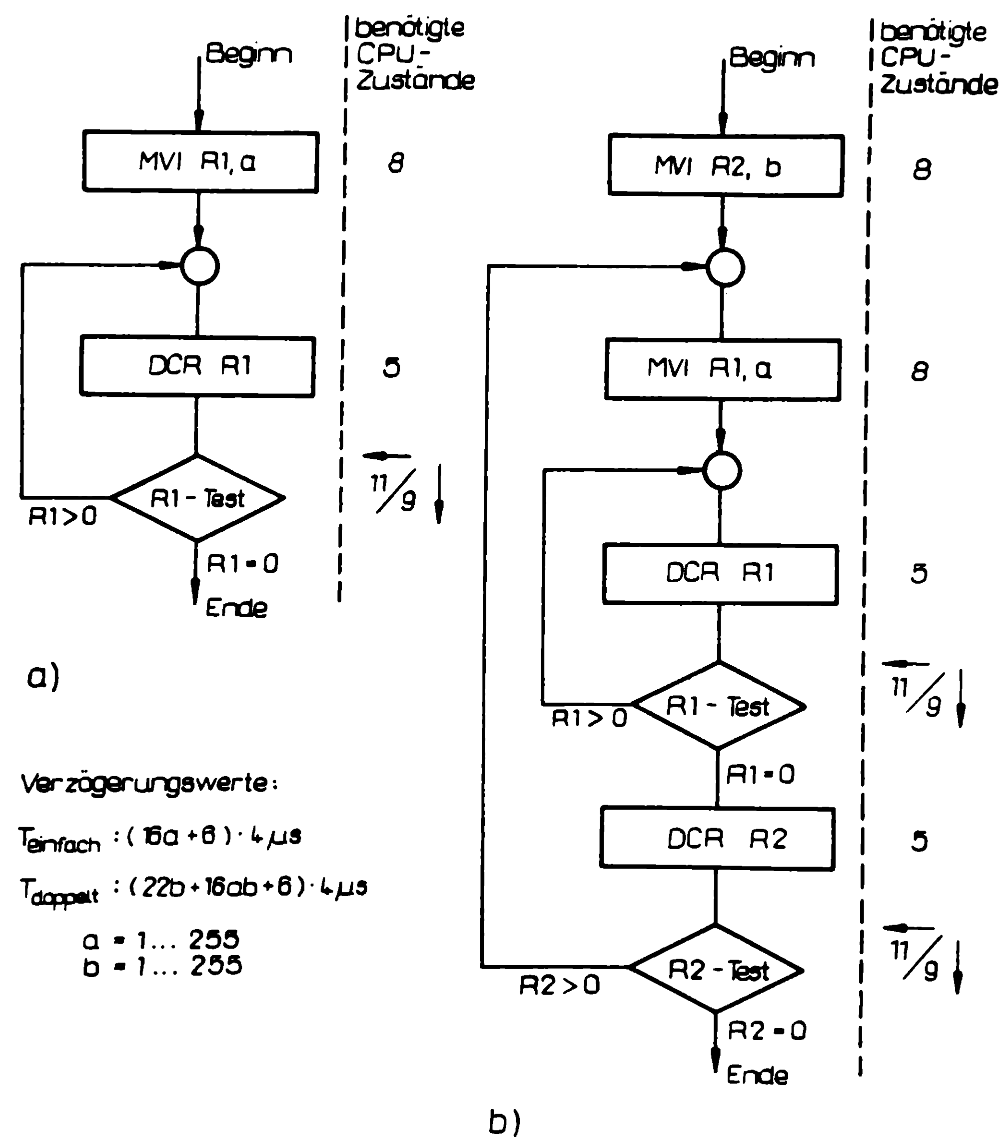


Bild 4.6: a) Einfache und b) doppelte Verzögerungsschleife (R1 und R2 sind prozessorinterne Register

Die Zählfunktion übernehmen prozessorinterne Register, die von einem bestimmten Ausgangswert an niedergezählt werden.
Mit der einfachen Schleife lassen sich maximal 16,34 Millisekunden und mit der doppelten Schleife maximal 4,184 Sekunden Verzögerung erzeugen, wenn 0,5 MHz Taktfrequenz zugrunde gelegt wird. Kürzere Verzögerungswerte lassen sich aus den abgeleiteten Beziehungen

$$T_{einfach} = (16 \cdot a + 6) \cdot 4\ \mu s$$

und

$$T_{doppelt} = (22 \cdot b + 16 \cdot a \cdot b + 6) \cdot 4\ \mu s$$

berechnen.
Für eine 1 ms-Verzögerung wählt man z. B. eine einfache Schleife mit $a = 15$, wobei sich ein Grundverzögerungswert von 984 µs ergibt. Durch Anketten von NOP-Befehlen (20 µs-Verzögerungsbefehle) an das Ende der Schleife können die Grundverzögerungswerte dem gewünschten Wert recht nahe gebracht werden.

4.5. Interruptprogrammierung

Zu der im Punkt 2.7. vorgestellten Interrupt-Baugruppe sollen hier die notwendige programmtechnischen Aspekte behandelt werden. Wird der MRS 1 mit mehreren Interrupts betrieben so ergeben sich zwei mögliche Betriebsarten der Interrupt-Baugruppe:

A) Die einzelnen Interrupts haben garantierte zeitliche Abstände voneinander die länger sind, als die längste Interrupt-Service-Routine (ISR).
In diesem Fall kann der Eingang INT-INH der Interrupt-Baugruppe fest auf H gelegt werden, und die Interrupts werden in der Reihenfolge ihrer Anmeldung verarbeitet.

B) Die einzelnen Interrupts können zu beliebigen Zeitpunkten eintreffen. Da die Interrupt-Baugruppe ein Unterbrechen einer ISR durch weitere Interrupts nicht zuläßt, werden die Anmeldungen bis zum Ende der gerade in Abarbeitung befindlichen ISR zwischengespeichert. Am Ende der ISR muß durch das Programm eine ENDE-Mitteilung an die Interrupt-Baugruppe erfolgen. Dies geschieht am Ende jeder ISR durch das 0/1-Setzen eines Bits eines Ausgabe-Tores, wobei dieser Ausgang mit dem Eingang $\overline{MSK}$

der Interrupt-Baugruppe verbinden sein muß.

Die ISR's für Interrupts nach Variante A werden durch einen Restart-Befehl (RST) begonnen und enden mit einem Return-Befehl oder auch mit einem Halt-Befehl (Bild 4.7).

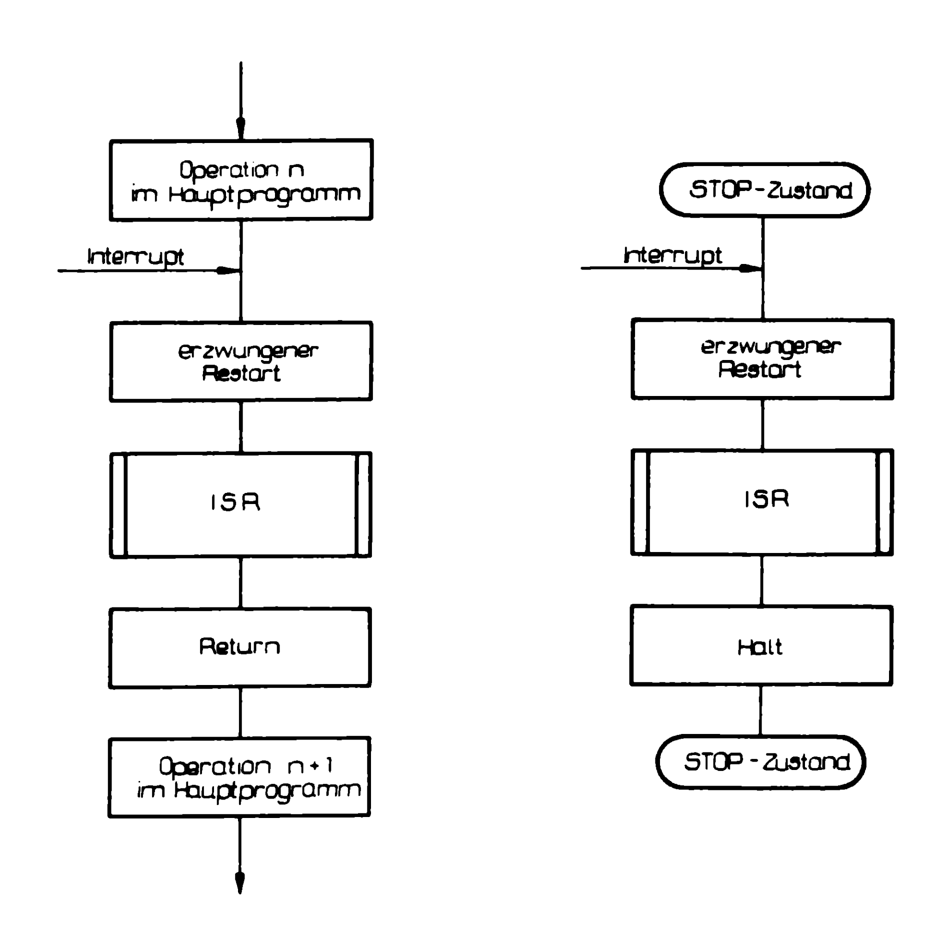


Bild 4.7: Möglichkeiten der Interruptbedienung für Interrupts nach Variante A

Bei Interrupts nach Variante B wird jede ISR ebenfalls mit einem Restart begonnen, vor Beendigung mittels Return- oder Halt-Befehlen muß jedoch zusätzlich die ENDE-Mitteilung vorbereitet und ausgegeben werden. Da der Eingang $\overline{MSK}$ der Interrupt-Baugruppe aus der 0 --- 1 Flanke die ENDE-Mitteilung entnimmt, genügt es, das Bit auf 0 und danach auf 1 zu setzen (Bild 4.8).

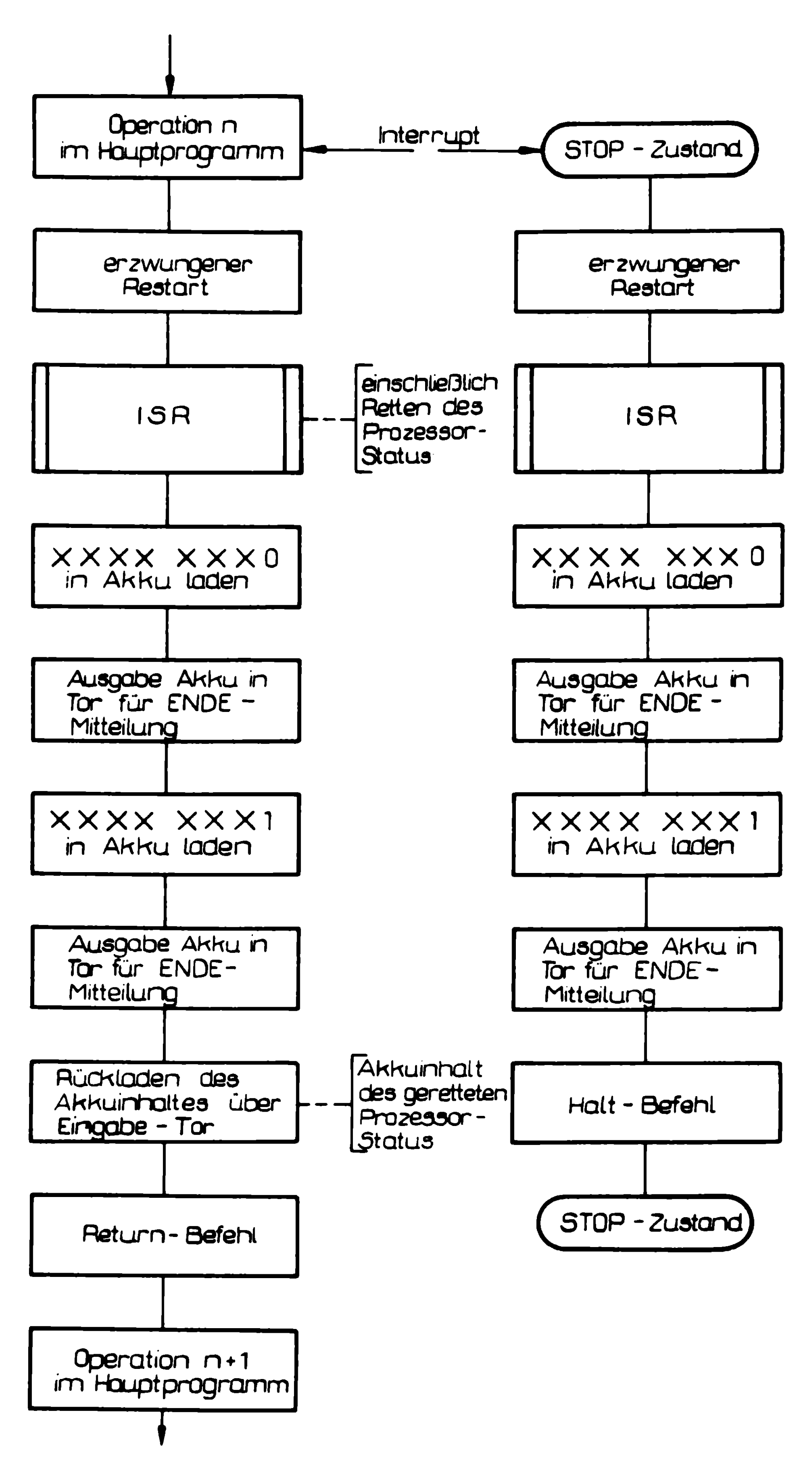


Bild 4.8: Möglichkeiten der Interruptbedienung für Interrupts der Variante B
(XXXX XXX = beliebige Belegung)

A 0. Begriffserklärung

Begriff	Erklärung
access	Zugriff; Möglichkeit, eine bestimmte Speicherzelle anzusprechen (zu adressieren)
address	Adresse; Bezeichnungskode für einen Speicherplatz
A/D-Wandler	Analog-Digital-Wandler; Baugruppe, die analoge Größen im Digitalkode ausdrückt
Akkumulator	Register mit der zusätzlichen Möglichkeit der binären Addition
ALU	arithmetic/logical unit; Teil einer Zentraleinheit, in der arithmetische und logische Operationen ausgeführt werden
Anwenderprogramm	Spezialprogramm, das zur Lösung eines Anwenderproblems dient (Gegensatz: Universal- oder Betriebsprogramm)
Befehl	Anweisung an Rechner zur Ausführung einer Aktion
Befehlsabarbeitungszyklus	Zyklus aus 3 bis 5 Zuständen
BCD-Ziffer	Binary Coded Decimal; in 4 bit (Binärkode) kodierte Ziffer
Bit	Binary Digit; binäre Informationseinheit
Bus	Sammelleitung; zur Übertragung von Daten, Adressen oder Signalen
Byte	kleinste adressierbare Dateneinheit
Carry	Übertrag
Compiler	Programm, welches Befehle einer höheren Programmsprache in Maschinenkode übersetzt
CPU	Central Processing Unit; Rechen- und Steuerwerk eines Rechnersystems

Begriff	Erklärung
Daten	kodierte Vorgänge oder Sachverhalte, speziell: Ein-/Ausgabeinformationen eines Rechnersystems
Datenbus	Sammelleitung zur Datenübertragung, an die mehrere Datengeräte angeschlossen sind
Datenspeicher	Speicher, in dem ausschließlich Daten gespeichert sind; speziell: meist RAM- in Ausnahmen auch ROM-Speicher
Debugprogramm	Programm, das zur Testung und Fehlersuche dient
DIL-Gehäuse	Dual-In-Line; Schaltkreisgehäuseform
DMA	Direct Memory Access; direkter Speicherzugriff; direkter, deshalb schneller Zugriff auf die Speicherbaugruppe (Datenspeicher) unter Umgehung der Zentraleinheit
EPROM	Erasable Programmable ROM; programmierbarer Festwertspeicher, dessen gesamter Informationsgehalt mittels UV-Licht löschbar ist
Flag	Flagge; Zustandsanzeige - Flip-Flop
Floating	Ausgangsstufen eines Bussenders befinden sich im hochohmigen Zustand
High	Logikpegel
Hardware	Sammelbegriff für Bauteile und Geräte
Interrupt	Unterbrechung des momentan in Arbeit befindlichen Programms und Bearbeitung einer ISR
ISR	Interrupt Service Routine; Unterprogramm, welches im Falle eines Interrupts bearbeitet wird
LIFO	Last In First Out; Arbeitsprinzip des Stackspeichers
Memory	Speicher; Medium, in dem Informationen über längere Zeit verfügbar gehalten werden

Begriff	Erklärung
Mikroprozessor	enthält Rechen- und Steuerwerk eines Mikrorechners
Mnemonic Code	alphanumerische Kurzbezeichnung für Befehle
MSB	Most Significant Bit; höchstwertigstes Bit eines Datenwortes
Operation Code	(Op Code); Maschinenbefehl in binärer Form
Polling	zyklisches Abfragen von Eingabestationen, langsamere Alternative zum Interrupt
Priorität	Wertigkeit
Programm	Folge von Befehlen
Programmspeicher	Speicher, in dem nur Programme gespeichert sind [ROM]
Programmzähler	Register, das die Speicherplatzadresse des nächstfolgenden Befehls angibt
PROM	Programmable ROM; programmierbarer Festwertspeicherbaustein
RAM	Random Access Memory; Schreib-Lese-Speicher mit wahlfreiem Zugriff
Refresh	Auffrischen; bei dynamischen RAM zyklischer Adressiervorgang zum Ausgleich von Ladungsverlusten durch Leckströme
Register	kleine, schnelle Zwischenspeicher
ROM	Read Only Memory; Festwertspeicher mit wahlfreiem Zugriff
Routine	Abk. für "Subroutine", Unterprogramm
Sign	Vorzeichen
Simulatorprogramm	Testprogramm, das den Befehlssatz des Mikrorechners im größeren Rechner (Wirtsrechner) simuliert
Software	Sammelbegriff für alle Arten von Programmen

Begriff	Erklärung
Stack	Stapelspeicher; Adreßregister, speichert bis zu 7 Rückkehradressen bei Aufruf von ISR
Stack-Pointer	Zähler, der den Füllstand des Stack's anzeigt
Zustand	Teil eines Befehlsabarbeitungszyklus
ZVE	Zentrale Versorgungseinheit; siehe CPU

Tafel A 1.1: Basisbefehlssatz des U 808 D

Befehlsliste:

Befehls-kurz-zeichen	Anzahl der Zeit-zustände	Befehlskode								Wirkung
		D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
Registerbefehle										
MOV r_1,r_2 1)	5	1	1	D	D	D	S	S	S	Lade Register r_1 mit dem Inhalt von Register r_2 ($r_1=r_2$: Leerbefehl)
MOV r,M 2)	8	1	1	D	D	D	1	1	1	Lade Register r mit dem Inhalt des Speicherplatzes, der durch den Inhalt der Register H und L adressiert wird
MVI r 3)	8	0 B_2	0 B_2	D B_2	D B_2	D B_2	1 B_2	1 B_2	0 B_2	Lade Register r mit den im 2. Byte des Befehls abgespeicherten Daten $B_2 \ldots B_2$
MOV M,r	7	1	1	1	1	1	S	S	S	Lade den Speicherplatz, der durch den Inhalt der Register H und L adressiert wird, mit dem Inhalt des Registers r
MVI M	9	0 B_2	0 B_2	1 B_2	1 B_2	1 B_2	1 B_2	1 B_2	0 B_2	Lade den Speicherplatz, der durch den Inhalt der Register H und L adressiert wird, mit den im 2. Byte des Befehls abgespeicherten Daten $B_2 \ldots B_2$
Die Ladebefehle beeinflussen nicht die Bedingungs-Flip-Flops										
INR r	5	0	0	D	D	D	0	0	0	Erhöhe den Inhalt des Registers r um 1 ($r \neq A$)
DCR r	5	0	0	D	D	D	0	0	1	Vermindere den Inhalt des Registers r um 1 ($r \neq A$)

Befehls-kurz-zeichen	Anzahl der Zeit-zustände	Befehlskode D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Wirkung
Die Zählbefehle beeinflussen alle Bedingungs-Flip-Flops, außer den Übertrags-Flip-Flop C.			
Haltbefehl:			
HLT HLT	4 4	0 0 0 0 0 0 0 X 1 1 1 1 1 1 1 1	Einnahme des Stop-zustandes bis zum nächsten Interrupt
Akkumulatorbefehle:			
ADD r	5	1 0 0 0 0 S S S	Addiere den Akkumulatorinhalt und den Inhalt des Registers r
ADD M	8	1 0 0 0 0 1 1 1	Addiere den Akkumulatorinhalt und den Inhalt des Speicherplatzes, der durch den Inhalt der Register H und L adressiert wird.
ADI	8	0 0 0 0 0 1 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2	Addiere den Akkumulatorinhalt und die im 2. Byte des Befehls abgespeicherten Daten $B_2 \ldots B_2$
ADC r	5	1 0 0 0 1 S S S	Addiere den Akkumulatorinhalt, den Übertrag und den Inhalt des Registers r.
ADC M	8	1 0 0 0 1 1 1 1	Addiere den Akkumulatorinhalt, den Übertrag und den Inhalt des Speicherplatzes, der durch den Inhalt der Register H und L adressiert wird.
ACI	8	0 0 0 0 1 1 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2	Addiere den Akkumulatorinhalt, den Übertrag und die im 2. Byte des Befehls abgespeicherten Daten $B_2 \ldots B_2$

Befehls-kurzzeichen	Anzahl der Zeitzustände	Befehlskode D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Wirkung
SUB r	5	1 0 0 1 0 S S S	Subtrahiere vom Akkumulatorinhalt des Registers r
SUB M	8	1 0 0 1 0 1 1 1	Subtrahiere vom Akkumulatorinhalt den Inhalt des Speicherplatzes, der durch den Inhalt der Register H und L adressiert wird.
SUI	8	0 0 0 1 0 1 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2	Subtrahiere vom Akkumulatorinhalt die im 2. Byte des Befehls abgespeicherten Daten $B_2 \ldots B_2$
SBB r	5	1 0 0 1 1 S S S	Subtrahiere vom Akkumulatorinhalt den Inhalt des Registers r
SBB M	8	1 0 0 1 1 1 1 1	Subtrahiere vom Akkumulatorinhalt den Übertrag und den Inhalt des Speicherplatzes, der durch den Inhalt der Register H und L adressiert wird.
SBI	8	0 0 0 1 1 1 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2	Subtrahiere vom Akkumulatorinhalt den Übertrag und die im 2. Byte des Befehls abgespeicherten Daten $B_2 \ldots B_2$
ANA r	5	1 0 1 0 0 S S S	Bilde das logische UND aus dem Akkumulatorinhalt und dem Register r

Befehls-kurz-zeichen	Anzahl der Zeit-zustände	Befehlskode D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Wirkung
ANA M	8	1 0 1 0 0 1 1 1	Bilde das logische UND aus dem Akkumulatorinhalt und dem Inhalt des Speicherplatzes, der durch den Inhalt der Register H und L adressiert wird
ANI	8	0 0 1 0 0 1 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2	Bilde das logische UND aus dem Akkumulatorinhalt und dem im 2. Byte des Befehls abgespeicherten Daten $B_2 \dots B_2$
XRA r	5	1 0 1 0 1 S S S	Bilde das Exclusiv-ODER aus dem Akkumulatorinhalt und dem Inhalt des Registers r
XRA M	8	1 0 1 0 1 1 1 1	Bilde das Exclusiv-ODER aus dem Akkumulatorinhalt und dem Inhalt des Speicherplatzes, der durch den Inhalt der Register H und L adressiert wird
XRI	8	0 0 1 0 1 1 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2	Bilde das Exclusiv-ODER aus dem Akkumulatorinhalt und den im 2. Byte des Befehls abgespeicherten Daten $B_2 \dots B_2$
ORA r	5	1 0 1 1 0 S S S	Bilde das Inclusiv-ODER aus dem Akkumulatorinhalt und dem Inhalt des Registers r

Befehls-kurz-zeichen	Anzahl der Zeit-zustände	Befehlskode D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Wirkung
ORA M	8	1 0 1 1 0 1 1 1	Bilde das Inclusiv-ODER aus dem Akkumulatorinhalt und dem Inhalt des Speicherplatzes, der durch den Inhalt der Register H und L adressiert wird
ORI	8	0 0 1 1 0 1 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2	Bilde das Inclusiv-ODER aus dem Akkumulatorinhalt und den im 2. Byte des Befehls abgespeicherten Daten $B_2 \ldots B_2$

Befehls-kurz-zeichen	Anzahl der Zeit-zustände	Befehlskode D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Wirkung
CMP r	5	1 0 1 1 1 S S S	Vergleich: Subtrahiere vom Akkumulatorinhalt den Inhalt des Registers r
CMP M	8	1 0 1 1 1 1 1 1	Vergleich: Subtrahiere vom Akkumulatorinhalt den Inhalt des Speicherplatzes, der durch den Inhalt der Register H und L adressiert wird
CPI	8	0 0 1 1 1 1 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2	Vergleich: Subtrahiere vom Akkumulatorinhalt die im 2. Byte des Befehls abgespeicherten Daten $B_2 \ldots B_2$

Der Akkumulatorinhalt wird beim Vergleich nicht geändert!

Die Ergebnisse der arithmetischen und logischen Befehle werden im Akkumulator abgespeichert (außer beim Vergleich!) und beeinflussen alle Bedingungs-Flip-Flops. Bei den logischen Befehlen wird der Übertragungs-Flip-Flop C Null gesetzt.

Befehls- kurz- zeichen	Anzahl der Zeit- zustände	Befehlskode D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Wirkung
RLC	5	0	0	0	0	0	0	1	0	Zyklische Links- verschiebung des Akkumulatorinhalts um 1 bit $A_{m+1} \leftarrow A_m$, $A_0 \leftarrow A_7$, $C \leftarrow A_7$
RRC	5	0	0	0	0	1	0	1	0	Zyklische Rechts- verschiebung des Akkumulatorinhalts um 1 bit $A_m \leftarrow A_{m+1}$, $A_7 \leftarrow A_0$, $C \leftarrow A_0$
RAL	5	0	0	0	1	0	0	1	0	Zyklische Links- verschiebung des Akkumulatorinhalts über den Übertrags- Flip-Flop C um 1 bit $A_{m+1} \leftarrow A_m$, $A_0 \leftarrow C$, $C \leftarrow A_7$
RAR	5	0	0	0	1	1	0	1	0	Zyklische Rechts- verschiebung des Akkumulatorinhalts über den Übertrags- Flip-Flop C um 1 bit $A_m \leftarrow A_{m+1}$, $A_7 \leftarrow C$, $C \leftarrow A_0$

Die Verschiebebefehle beeinflussen nur den Übertrags-Flip-Flop C.

Adressenoperationen: JUMP-Befehle

JMP	11	0	1	X	X	X	1	0	0	unbedingter Sprung zu der im 2. und 3. Byte des Befehls angegebenen Adresse
4)		B_2	B_2	B_2	B_2	B_2	B_2	B_2	B_2	
5)		X	X	B_3	B_3	B_3	B_3	B_3	B_3	
	9 oder 11									Sprung zu der im 2. und 3. Byte des Befehls angegebenen Adresse, wenn das ausgewählte Bedingungs-Flip-Flop Null ist

Befehls- kurz- zeichen	Anzahl der Zeit- zustände	Befehlskode D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Wirkung
		C_4 C_3	
JNC 6)		0 1 0 0 0 0 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	C = 0
JNZ		0 1 0 0 1 0 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	Z = 0
JP		0 1 0 1 0 0 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	S = 0
JPO		0 1 0 1 1 0 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	P = 0
	9 oder 11		Sprung zu der im 2. und 3. Byte des Befehls angegebenen Adresse, wenn das ausgewählte Bedingungs-Flip-Flop Eins ist
		C_4 C_3	
JC		0 1 1 0 0 0 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	C = 1
JZ		0 1 1 0 1 0 0 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	Z = 1

Befehls-kurz-zeichen	Anzahl der Zeit-zustände	Befehlskode D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Wirkung
JM		0	1	1	1	0	0	0	0	
		B_2	B_2	B_2	B_2	B_2	B_2	B_2	B_2	S = 1
		X	X	B_3	B_3	B_3	B_3	B_3	B_3	
JPE		0	1	1	1	1	0	0	0	
		B_2	B_2	B_2	B_2	B_2	B_2	B_2	B_2	P = 1
		X	X	B_3	B_3	B_3	B_3	B_3	B_3	
					CALL-Befehle					
CALL	11	0	1	X	X	X	1	1	0	unbedingter Aufruf des Unterprogramms auf der im 2. und 3. Byte des Befehls angegebenen Adresse
		B_2	B_2	B_2	B_2	B_2	B_2	B_2	B_2	
		X	X	B_3	B_3	B_3	B_3	B_3	B_3	
	9 oder 11									Aufruf des Unterprogramms auf der im 2. und 3. Byte des Befehls angegebenen Adresse, wenn der ausgewählte Bedingungs-Flip-Flop Null ist
					C_4	C_3				
CNC		0	1	0	0	0	0	1	0	
		B_2	B_2	B_2	B_2	B_2	B_2	B_2	B_2	C = 0
		X	X	B_3	B_3	B_3	B_3	B_3	B_3	
CNZ		0	1	0	0	1	0	1	0	
		B_2	B_2	B_2	B_2	B_2	B_2	B_2	B_2	Z = 0
		X	X	B_3	B_3	B_3	B_3	B_3	B_3	
CP		0	1	0	1	0	0	1	0	
		B_2	B_2	B_2	B_2	B_2	B_2	B_2	B_2	S = 0
		X	X	B_3	B_3	B_3	B_3	B_3	B_3	
CPO		0	1	0	1	1	0	1	0	
		B_2	B_2	B_2	B_2	B_2	B_2	B_2	B_2	P = 0
		X	X	B_3	B_3	B_3	B_3	B_3	B_3	

Befehls-kurz-zeichen	Anzahl der Zeit-zustände	Befehlskode D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Wirkung
	9 oder 11		Aufruf des Unter-programms auf der im 2. und 3. Byte des Befehls ange-gebenen Adresse, wenn der ausge-wählte Bedingungs-Flip-Flop Eins ist
		C_4 C_3	
CC		0 1 1 0 0 0 1 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	C = 1
CZ		0 1 1 0 1 0 1 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	Z = 1
CM		0 1 1 1 0 0 1 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	S = 1
CPE		0 1 1 1 1 0 1 0 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	P = 1
Die letzte Adresse des rufenden Programms wird im Adressen-speicher konserviert.			
		RETURN-Befehle	
RET	5	0 0 X X X 1 1 1	unbedingter Rück-sprung zur letzten Adresse im Adressenspeicher
	3 oder 5		Rücksprung zur letzten Adresse im Adressenspeicher, wenn der ausge-wählte Bedingungs-Flip-Flop Null ist

Befehlskurzzeichen	Anzahl der Zeitzustände	Befehlskode D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Wirkung
		C_4 C_3	
RNC		0 0 0 0 0 0 1 1 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	C = 0
RNZ		0 0 0 0 1 0 1 1 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	Z = 0
RP		0 0 0 1 0 0 1 1 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	S = 0
RPO		0 0 0 1 1 0 1 1 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	P = 0
	3 oder 5		Rücksprung zur letzten Adresse im Adressenspeicher, wenn der ausgewählte Bedingungs-Flip-Flop Eins ist
		C_4 C_3	
RC		0 0 1 0 0 0 1 1 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	C = 1
RZ		0 0 1 0 1 0 1 1 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	Z = 1
RM		0 0 1 1 0 0 1 1 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	S = 1

Befehls-kurz-zeichen	Anzahl der Zeit-zustände	Befehlskode D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0	Wirkung
RPE		0 0 1 1 1 0 1 1 B_2 B_2 B_2 B_2 B_2 B_2 B_2 B_2 X X B_3 B_3 B_3 B_3 B_3 B_3	P = 1
Ist bei den bedingten Adressenoperationen die Sprungbedingung nicht erfüllt, wird der folgende Befehl im Speicher adressiert und es werden weniger Zeitzustände benötigt (9 bzw. 3).			
RST	5	0 0 A A A 1 0 1	Unbedingter Aufruf des Unterprogramms ab Speicheradresse AAAØØØ (AAA ≙ $D_5D_4D_3$).
Die letzte Adresse des rufenden Programms wird im Adressenspeicher konserviert.			
Eingabe-/Ausgabe-Befehle:			
IN	8	0 1 0 0 M M M 1	Eingabe des Inhalts des ausgewählten Eingabetores MMM in den Akkumulator
OUT	6	0 1 R R M M M 1	Ausgabe des Inhalts des Akkumulators in das ausgewählte Ausgaberegister RRMMM (RR ≠ 0)

1) SSS = Quell-Indexregister r_2
   DDD = Bestimmungs-Indexregister r_1
   Die Indexregister sind bezeichnet mit A(kkumulator-000), B (001), C (010), D (011), E (100), H (101), L (110)

2) Das Memoryregister wird adressiert durch den Inhalt der Indexregister H und L

3) Das zusätzliche Datenwort des Befehls wird bezeichnet mit $B_2 \ldots B_2$

4) X = nicht benötigte Bits

5) Die zusätzlichen Adresswörter des Befehls werden bezeichnet mit

$$B_2 \ldots B_2$$
$$XX\ B_3 \ldots B_3$$

6) Die Bedingungs-Flip-Flops sind definiert durch C_4C_3:
   C (00-Übertrag), Z (01-Ergebnis hat Nullinhalt), S (10-MSB vom Ergebnis ist 1), P (11-Parität ist vorhanden)

Tafel A 1.2.: Ausführlicher Befehlssatz des U 808 D

Tafel A 1.2.1.: Einzelregister-Befehle

Befehlsformat: KODE REGM
Bemerkung: REGM $\neq$ A oder M

KODE		Beschreibung
INR	(REGM) ⟵ (REGM) + 1	Erhöhte Register REGM
DCR	(REGM) ⟵ (REGM) - 1	Erniedrige Register REGM

Bedingungs-Flip-Flops Z, S und P werden beeinflußt

Tafel A 1.2.2.: MOV-Befehle

Befehlsformat: MOV DST, SRC
Bemerkung: SRC UND DST nicht gleichzeitig M

KODE		Beschreibung
MOV	(DST) ⟵ (SRC)	Lade Register DST von Register SRC

Bedingungs-Flip-Flops werden nicht beeinflußt

Tafel A 1.2.3.: Register oder Memory zu Akkumulator-Befehle

Befehlsformat: KODE REGM

KODE		Beschreibungen
ADD	(A)⟵(A) + (REGM)	Addition von REGM und Akkumulator
ADC	(A)⟵(A) + (REGM) + (Übertrag)	Addition von REGM Übertrag und Akkumulator
SUB	(A)⟵(A) - (REGM)	Subtraktion von REGM und Akkumulator
SBB	(A)⟵(A) - (REGM) - (Übertrag)	Subtraktion von REGM, Übertrag und Akkumulator
ANA	(A)⟵(A) AND (REGM)	AND zwischen REGM und Akkumulator
XRA	(A)⟵(A) XOR (REGM)	Exklusives ODER zwischen REGM und Akkumulator
ORA	(A)⟵(A) OR (REGM)	Inklusives ODER zwischen REGM und Akkumulator
CMP	Bedingungs-Flip-Flops werden gesetzt durch (A) - (REGM)	Vergleich zwischen REGM und Akkumulator

Bedingungs-Flip-Flops werden beeinflußt zu:

ADD, ADC, SUB, SBB:	C, S, Z, P	
ANA, XRA, ORA:	S, Z, P;	C wird zu null
CMP:	C, S, Z, P,	Z wird gesetzt, wenn (A) = (REGM) C wird zurückgesetzt, wenn (A) ≥ (REGM) C wird gesetzt, wenn (A) < (REGM)

Tafel A 1.2.4.: Akkumulator-Rotations-Befehle

Befehlsformat: KODE

KODE		Beschreibung
RLC	(Übertrag) $\leftarrow A_7$ $A_{n+1} \leftarrow A_n$, $A_0 \leftarrow A_7$	Übertrag wird gesetzt von A_7, Akkumulatorverschiebung nach links
RRC	(Übertrag) $\leftarrow A_0$ $A_n \leftarrow A_{n+1}$, $A_7 \leftarrow A_0$	Übertrag wird gesetzt von A_0, Akkumulatorverschiebung nach rechts
RAL	$A_{n+1} \leftarrow A_n$ (Übertrag) $\leftarrow A_7$, $A_0 \leftarrow$ (Übertrag)	Verschiebung des Akkumulators nach links durch den Übertrag
RAR	$A_n \leftarrow A_{n+1}$, (Übertrag) $\leftarrow A_0$, $A_7 \leftarrow$ (Übertrag)	Verschiebung des Akkumulators nach rechts durch den Übertrag

Bedingungs-Flip-Flop C wird beeinflußt

Tafel A 1.2.5.: Unmittelbare DATEN-Befehle

Befehlsformat: MVI REGM, DATEN
oder
KODE DATEN

KODE		Beschreibung
MVI	(REGM) ← DATEN	Lade REGM unmittelbar mit den DATEN
ADI	(A) ← (A) + DATEN	Addition von DATEN und Akkumulator
ACI	(A) ← (A) + DATEN + (Übertrag)	Addition von DATEN, Übertrag und Akkumulator
SUI	(A) ← (A) - DATEN	Subtraktion von DATEN und Akkumulator
SBI	(A) ← (A) - DATEN - (Übertrag)	Subtraktion von DATEN, Übertrag und Akkumulator
ANI	(A) ← (A) AND DATEN	AND zwischen DATEN und Akkumulator
XRI	(A) ← (A) XOR DATEN	exklusives ODER zwischen DATEN und Akkumulator
ORI	(A) ← (A) OR DATEN	Inklusives ODER zwischen DATEN und Akkumulator
CPI	Bedingungs-Flip-Flops werden gesetzt durch (A) - DATEN	Vergleich zwischen DATEN und Akkumulator

Bedingungs-Flip-Flops werden beeinflußt zu:

MVI:	keine Beeinflussung
ADI, ACI, SUI, SBI:	C, S, Z, P
ANI, XRI, ORI:	S, Z, P; C wird null
CMP:	C, S, Z, P
	Z wird gesetzt, wenn (A) = (REGM)
	C wird rückgesetzt, wenn (A) ≥ (REGM)
	C wird gesetzt, wenn (A) < (REGM)

Tafel A 1.2.6.: Sprung-Befehle

Befehlsformat: KODE ADR

KODE		Beschreibung
JMP	(PC)←ADR	Sprung auf ADR
JC	wenn C = 1, (PC)←ADR wenn C = 0, (PC)←(PC) + 3	Sprung auf ADR, wenn Übertrags-Flip-Flop gesetzt ist
JNC	wenn C = 0, (PC)←ADR wenn C = 1, (PC)←(PC) + 3	Sprung auf ADR, wenn Übertrags-Flip-Flop rückgesetzt ist
JZ	wenn Z = 1, (PC)←ADR wenn Z = 0, (PC)←(PC) + 3	Sprung auf ADR, wenn Null-Flip-Flop gesetzt ist
JNZ	wenn Z = 0, (PC)←ADR wenn Z = 1, (PC)←(PC) + 3	Sprung auf ADR, wenn Null-Flip-Flop rück-gesetzt ist
JM	wenn S = 1, (PC)←ADR wenn S = 0, (PC)←(PC) + 3	Sprung auf ADR, wenn MSB-Flip-Flop gesetzt ist
JP	wenn S = 0, (PC)←ADR wenn S = 1, (PC)←(PC) + 3	Sprung auf ADR, wenn MSB-Flip-Flop rück-gesetzt ist
JPE	wenn P = 1, (PC)←ADR wenn P = 0, (PC)←(PC) + 3	Sprung auf ADR, wenn Paritäts-Flip-Flop gesetzt ist
JPO	wenn P = 0, (PC)←ADR wenn P = 1, (PC)←(PC) + 3	Sprung auf ADR, wenn Paritäs-Flip-Flop rückgesetzt ist

Bedingungs-Flip-Flops werden nicht beeinflußt

Tafel A 1.2.7.: Ruf-Befehle

Befehlsformat: KODE ADR

KODE		Beschreibung
CALL	(STK) ⟵ (PC), (PC) ⟵ (ADR)	Ruf der Unterroutine und Retten der Rückkehradresse im Stack
CC	wenn C = 1, (STK) ⟵ (PC), (PC) ⟵ (ADR) wenn C = 0, (PC) ⟵ (PC) + 3	Ruf der Unterroutine, wenn Übertrags-Flip-Flop gesetzt ist
CNC	wenn C = 0, (STK) ⟵ (PC), (PC) ⟵ (ADR) wenn C = 1, (PC) ⟵ (PC) + 3	Ruf der Unterroutine, wenn Übertrags-Flip-Flop rückgesetzt ist
CZ	wenn Z = 1, (STK) ⟵ (PC), (PC) ⟵ (ADR) wenn Z = 0, (PC) ⟵ (PC) + 3	Ruf der Unterroutine, wenn Null-Flip-Flop gesetzt ist
CNZ	wenn Z = 0, (STK) ⟵ (PC) (PC) ⟵ (ADR) wenn Z = 1, (PC) ⟵ (PC) + 3	Ruf der Unterroutine, wenn Null-Flip-Flop rückgesetzt ist
CM	wenn S = 1, (STK) ⟵ (PC) (PC) ⟵ (ADR) wenn S = 0, (PC) ⟵ (PC) + 3	Ruf der Unterroutine, wenn MSB-Flip-Flop gesetzt ist
CP	wenn S = 0, (STK) ⟵ (PC) (PC) ⟵ (ADR) wenn S = 1, (PC) ⟵ (PC) + 3	Ruf der Unterroutine, wenn MSB-Flip-Flop rückgesetzt ist

CPE	wenn P = 1, (STK) ⟵ (PC), (PC) ⟵ (ADR) wenn P = 0, (PC) ⟵ (PC) + 3	Ruf der Unterroutine, wenn Paritäts-Flip-Flop gesetzt ist
CPO	wenn P = 0 (STK) ⟵ (PC), (PC) ⟵ (ADR) wenn P = 1 (PC) ⟵ (PC) + 3	Ruf der Unterroutine wenn Paritäts-Flip-Flop rückgesetzt ist

---

Bedingungs-Flip-Flops werden nicht beeinflußt

Tafel A 1.2.8.: Rückkehr-Befehle

Befehlsformat: KODE KODE		Beschreibung
RET	(PC) ← (STK)	Rückkehr aus einer Unterroutine
RC	wenn C = 1, (PC) ← (STK) wenn C = 0, (PC) ← (PC) + 1	Rückkehr, wenn Übertrags-Flip-Flop gesetzt ist
RNC	wenn C = 0, (PC) ← (STK) wenn C = 1, (PC) ← (PC) + 1	Rückkehr, wenn Übertrags-Flip-Flop rückgesetzt ist
RZ	wenn Z = 1, (PC) ← (STK) wenn Z = 0, (PC) ← (PC) + 1	Rückkehr, wenn Null-Flip-Flop gesetzt ist
RNZ	wenn Z = 0, (PC) ← (STK) wenn Z = 1, (PC) ← (PC) + 1	Rückkehr, wenn Null-Flip-Flop rückgesetzt ist
RM	wenn S = 1, (PC) ← (STK) wenn S = 0, (PC) ← (PC) + 1	Rückkehr, wenn MSB-Flip-Flop gesetzt ist
RP	wenn S = 0, (PC) ← (STK) wenn S = 1, (PC) ← (PC) + 1	Rückkehr, wenn MSB-Flip-Flop rückgesetzt ist
RPE	wenn P = 1, (PC) ← (STK) wenn P = 0, (PC) ← (PC) + 1	Rückkehr, wenn Paritäts-Flip-Flop gesetzt ist
RPO	wenn P = 0, (PC) ← (STK) wenn P = 1, (PC) ← (PC) + 1	Rückkehr, wenn Paritäts-Flip-Flop rückgesetzt ist

Bedingungs-Flip-Flops werden nicht beeinflußt

Tafel A 1.2.9.: Restart-Befehle

Befehlsformat: RST EXP
Bemerkung: $0 \leqq EXP \leqq 7$

KODE		Beschreibung
RST	(STK) ← (PC) (PC) ← 00000000EXP000	Aufruf der Subroutine, die durch EXP spezifisch adressiert ist

Bedingungs-Flip-Flops werden nicht beeinflußt

Tafel A 1.2.10.: Input/Output-Befehle

Befehlsformat: KODE EXP
Bemerkung: für IN, $0 \leqq EXP \leqq 7$
für OUT, $8 \leqq EXP \leqq 31$

KODE		Beschreibung
IN	(A) ← (Eingabe-Tor)	Lesen eines Bytes vom Eingabe-Tor, das durch EXP spezifiziert ist, in den Akkumulator
OUT	(Ausgabe-Tor) ← (A)	Senden des Akkumulatorinhaltes in das Ausgabe-Tor, das durch EXP spezifiziert ist

Bedingungs-Flip-Flops werden nicht beeinflußt

Tafel A 1.3.: Korrespondenztafel Hexadezimal-Mnemoniccode

MOVE		ACCUMULATOR[1]	
----------	E8 MOV H,A	80 ADD A	A8 XRA A
C1 MOV A,B	E9 MOV H,B	81 ADD B	A9 XRA B
C2 MOV A,C	EA MOV H,C	82 ADD C	AA XRA C
C3 MOV A,D	EB MOV H,D	83 ADD D	AB XRA D
C4 MOV A,E	EC MOV H,E	84 ADD E	AC XRA E
C5 MOV A,H	ED MOV H,H	85 ADD H	AD XRA H
C6 MOV A,L	EE MOV H,L	86 ADD L	AE XRA L
C7 MOV A,M	EF MOV H,M	87 ADD M	AF XRA M
C8 MOV B,A	F0 MOV L,A	88 ADC A	B0 ORA A
C9 MOV B,B	F1 MOV L,B	89 ADC B	B1 ORA B
CA MOV B,C	F2 MOV L,C	8A ADC C	B2 ORA C
CB MOV B,D	F3 MOV L,D	8B ADC D	B3 ORA D
CC MOV B,E	F4 MOV L,E	8C ADC E	B4 ORA E
CD MOV B,H	F5 MOV L,H	8D ADC H	B5 ORA H
CE MOV B,L	F6 MOV L,L	8E ADC L	B6 ORA L
CF MOV B,M	F7 MOV L,M	8F ADC M	B7 ORA M
D0 MOV C,A	F8 MOV M,A	90 SUB A	B8 CMP A
D1 MOV C,B	F9 MOV M,B	91 SUB B	B9 CMP B
D2 MOV C,C	FA MOV M,C	92 SUB C	BA CMP C
D3 MOV C,D	FB MOV M,D	93 SUB D	BB CMP D
D4 MOV C,E	FC MOV M,E	94 SUB E	BC CMP E
D5 MOV C,H	FD MOV M,H	95 SUB H	BD CMP H
D6 MOV C,L	FE MOV M,L	96 SUB L	BE CMP L
D7 MOV C,M	----------	97 SUB M	BF CMP M
D8 MOV D,A		98 SBB A	
D9 MOV D,B		99 SBB B	
DA MOV D,C		9A SBB C	
DB MOV D,D		9B SBB D	
DC MOV D,E		9C SBB E	
DD MOV D,H		9D SBB H	
DE MOV D,L		9E SBB L	
DF MOV D,M		9F SBB M	
E0 MOV E,A		A0 ANA A	
E1 MOV E,B		A1 ANA B	
E2 MOV E,C		A2 ANA C	
E3 MOV E,D		A3 ANA D	
E4 MOV E,E		A4 ANA E	
E5 MOV E,H		A5 ANA H	
E6 MOV E,L		A6 ANA L	
E7 MOV E,M		A7 ANA M	

JUMP (Adr)

44	JMP
40	JNC
48	JNZ
50	JP
58	JPO
60	JC
68	JZ
70	JM
78	JPE

CALL (Adr)

46	CALL
42	CNC
4A	CNZ
52	CP
5A	CPO
62	CC
6A	CZ
72	CM
7A	CPE

RETURN

07	RET
03	RNC
0B	RNZ
13	RP
1B	RPO
23	RC
2B	RZ
33	RM
3B	RPE

RESTART

05	RST 0
0D	RST 1
15	RST 2
1D	RST 3
25	RST 4
2D	RST 5
35	RST 6
3D	RST 7

MOVE IMMEDIATE (D8)

06	MVI A,
0E	MVI B,
16	MVI C,
1E	MVI D,
26	MVI E,
2E	MVI H,
36	MVI L,
3E	MVI M,

Acc IMMEDIATE[1)] (D8)

04	ADI
0C	ACI
14	SUI
1C	SBI
24	ANI
2C	XRI
34	ORI
3C	CPI

INPUT

41	IN 0
43	IN 1
45	IN 2
47	IN 3
49	IN 4
4B	IN 5
4D	IN 6
4F	IN 7

INCREMENT[2)]

08	INR	B
10	INR	C
18	INR	D
20	INR	E
28	INR	H
30	INR	L

DECREMENT[2)]

09	DCR	B
11	DCR	C
19	DCR	D
21	DCR	E
29	DCR	H
31	DCR	L

OUTPUT

51	OUT 8
53	OUT 9
55	OUT 10
57	OUT 11
59	OUT 12
5B	OUT 13
5D	OUT 14
5F	OUT 15
61	OUT 16
63	OUT 17
65	OUT 18
67	OUT 19
69	OUT 20
6B	OUT 21
6D	OUT 22
6F	OUT 23
71	OUT 24
73	OUT 25
75	OUT 26
77	OUT 27
79	OUT 28
7B	OUT 29
7D	OUT 30
7F	OUT 31

ROTATE[3)]

02	RLC
0A	RRC
12	RAL
1A	RAR

CONTROL

00	HLT
C0	NOP

Adr = 14 bit Adresse
D8 = Konstante oder logischer/arithmetischer Ausdruck mit der Wortbreite von 8 bit
1) = Alle Bedingungs-Flip-Flops (C,Z,S,P) werden beeinflußt
2) = Alle Bedingungs-Flip-Flops (außer C) werden beeinflußt
3) = Nur Bedingungs-Flip-Flop C wird beeinflußt

Tafel A 1.4.: Arbeitstabelle für die Korrespondenz Maschinencode - Mnemoniccode

$D_0 \ldots D_3$ \ $D_4 \ldots D_7$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	HLT	INR C	INR E	INR L	JNC	JP	JC	JM	ADD A	SUB A	ANA A	ORA A	NOP	MOV C,A	MOV E,A	MOV L,A
1	-	DCR C	DCR E	DCR L	IN 0	OUT 8	OUT 16	OUT 24	ADD B	SUB B	ANA B	ORA B	MOV A,B	MOV C,B	MOV E,B	MOV L,B
2	RLC	RAL	-	-	CNC	CP	CC	CM	ADD C	SUB C	ANA C	ORA C	MOV A,C	MOV C,C	MOV E,C	MOV L,C
3	RNC	RP	RC	RM	IN 1	OUT 9	OUT 17	OUT 25	ADD D	SUB D	ANA D	ORA D	MOV A,D	MOV C,D	MOV E,D	MOV L,D
4	ADI	SUI	ANI	ORI	JMP	JMP	JMP	JMP	ADD E	SUB E	ANA E	ORA E	MOV A,E	MOV C,E	MOV E,E	MOV L,E
5	RST 0	RST 2	RST 4	RST 6	IN 2	OUT 10	OUT 18	OUT 26	ADD H	SUB H	ANA H	ORA H	MOV A,H	MOV C,H	MOV E,H	MOV L,H
6	MVI A	MVI C	MVI E	MVI L	CALL	CALL	CALL	CALL	ADD L	SUB L	ANA L	ORA L	MOV A,L	MOV C,L	MOV E,L	MOV L,L
7	RET	RET	RET	RET	IN 3	OUT 11	OUT 19	OUT 27	ADD M	SUB M	ANA M	ORA M	MOV A,M	MOV C,M	MOV E,M	MOV L,M
8	INR F	INR D	INR H	-	JNZ	JPO	JZ	JPE	ADC A	SBB A	XRA A	CMP A	MOV B,A	MOV D,A	MOV H,A	MOV M,A
9	DCR B	DCR D	DCR H	-	IN 4	OUT 12	OUT 20	OUT 28	ADC B	SBB B	XRA B	CMP B	MOV B,B	MOV D,B	MOV H,B	MOV M,B
A	RRC	RAR	-	-	CNZ	CPO	CZ	CPE	ADC C	SBB C	XRA C	CMP C	MOV B,C	MOV D,C	MOV H,C	MOV M,C
B	RNZ	RPO	RZ	RPE	IN 5	OUT 13	OUT 21	OUT 29	ADC D	SBB D	XRA D	CMP D	MOV B,D	MOV D,D	MOV H,D	MOV M,D
C	ACI	SBI	XRI	CPI	JMP	JMP	JMP	JMP	ADC E	SBB E	XRA E	CMP E	MOV B,E	MOV D,E	MOV H,E	MOV M,E
D	RST 1	RST 3	RST 5	RST 7	IN 6	OUT 14	OUT 22	OUT 30	ADC H	SBB H	XRA H	CMP H	MOV B,H	MOV D,H	MOV H,H	MOV M,H
E	MVI B	MVI D	MVI H	MVI M	CALL	CALL	CALL	CALL	ADC L	SBB L	XRA L	CMP L	MOV B,L	MOV D,L	MOV H,L	MOV M,L
F	RET	RET	RET	RET	IN 7	OUT 15	OUT 23	OUT 31	ADC M	SBB M	XRA M	CMP M	MOV B,M	MOV D,M	MOV H,M	HLT

A 2.0. Signalerklärungen

Tafel A 2.0.1. Signale auf Karte CPU 1

Signal	TTL-Standard-Lastfaktor F_I	F_O	Erläuterung
MDB	4		Memory Data Bus, Eingangsdatenbus von der Speicherbaugruppe
INB	4		Input Bus, Eingangsdatenbus von der Ein-/(Aus)gabebaugruppe
IIB	4		Interrupt Input Bus, Eingangsdatenbus von der Interruptbaugruppe
DB		7	Data Bus, Datenbus des µR
MAD'		7	Memory Address, Adreßbus des µR
MAD		7	Memory Address, Adreßbus zur Speicherbaugruppe des µR
$\overline{\text{DB IN}}$	4		DB Input, Übernahme des selektierten Eingangsdatenbusses in die CPU (L ist auslösender Pegel)
$\overline{\text{INT SEL}}$	4		Interrupt Select, selektiert Interruptzyklus
$\overline{\text{I/O IN SEL}}$	4		In/Out Input Select, selektiert IN/OUT-Operation
INT	3		Interrupt, Interruptmeldung an die CPU (H ist auslösender Pegel)
$\overline{\text{WAIT REQ}}$	3		Wait Request, WAIT-Anforderung an die CPU
cp	1		clockpulse, Taktimpulse
sync'		10	synchronisiert Taktablauf mit CPU-Betrieb
s		10	status, kodierte Zustandsinformation der CPU
$\overline{\text{T1}}$, $\overline{\text{T2}}$	2		dekodierte Zustandsinformationen
$\overline{\text{DB OUT}}$	2		DB Output, Ausgabe des µR-Datenbusses

Tafel A 2.0.2. Signale auf Karte CPU 2

Signal	TTL-Standard-Lastfaktoren F_I	F_O	Erläuterung
INT		10	Interrupt, siehe CPU 1
$\overline{T1I}$, $\overline{T1}$, $\overline{T2}$,		7	dekodierte Zustandsinformationen
$\overline{T3W}$, $\overline{T3}$, $\overline{T3S}$		6	
$\overline{I/O\ OUT}$		10	IN/OUT Output, Befehl zur Datenübernahme in selektiertes OUT-Port
MAD'	4		Memory Address, siehe CPU 1
$\overline{T3A}$		8	den Zustand T3 umfassendes Signal
$\overline{PCI}$		10	Instruction Fetch Cycle, zeigt Befehlsholzyklus an
$\overline{PCR}$		10	Memory Read Cycle, zeigt Datenlesezyklus an
$\overline{PCW}$		7	Memory Write Cycle, zeigt Datenschreibzyklus an
$\overline{PCC}$		8	Input/Output Cycle, zeigt Ein-/Ausgabezyklus an
cp		5	clock pulse, siehe CPU 1
s	5		status, siehe CPU 1
$\overline{I/O\ IN\ SEL}$		9	In/Out Input Select, siehe CPU 1
R/$\overline{W}$		10	Read/Write, schaltet RAM's von Lesen auf Schreiben
sync'	1		siehe CPU 1
$\overline{INT\ SEL}$		10	Interrupt Select, siehe CPU 1
$\overline{INT\ REQ}$	5		Interrupt Request, Interruptanforderung an CPU-Baugruppe
$\overline{HLT\ INT\ REQ}$		10	Halt Interrupt Request, Interruptanforderung, wird nur im Zustand T3S wirksam

Signal	TTL-Standard-Lastfaktoren		Erläuterung
	F_I	F_O	
$\overline{\text{INT REQ LTH}}$		10	Interrupt Request Latch, zeigt Empfang einer Interruptanforderung an
sync A		8	auf Taktflanke $\overline{\text{cp2}}$ bezogenes Signal sync'
$\overline{\text{CI}}$		10	Cycle Interrupt, zyklisches Interrupt-signal
$\overline{\text{CLE}}$		10	Clear Enable, Anlaufinterruptsignal
$\overline{\text{CLA}}$		1	Clear All, betriebsspannungsabhängiges Löschsignal
$\overline{\text{DB OUT}}$		10	DB Output, siehe CPU 1
$\overline{\text{DB IN}}$		10	DB Input, siehe CPU 1
$\overline{\text{HOLD ACK}}$		10	Hold Acknowledge, Rückmeldung für DMA-Betrieb
$\overline{\text{HOLD REQ}}$	4		Hold Request, Anforderung der DMA-Betriebsweise
$\overline{\text{WAIT REQ}}$	2		Wait Request, siehe CPU 1

Tafel A 2.0.3. Signale auf Karte RAM/ROM

Signal	TTL-Standard-Lastfaktoren		Erläuterung
	F_I	F_O	
COL SEL	2	10	Column Select, Spaltenansteuerung der Speichermatrix
LINE SEL	2	10	Line Select, Zeilenansteuerung der Speichermatrix
LINE SEL'	2		Line Select, Zeilenansteuerung, dient der Kartenspezifikation
CARD SEL	2	10	Card Select, Kartenansteuerung
MCS	2		Memory Card Select, dient der Kartenspezifikation
R/$\overline{\text{W}}$	1		Read/Write, siehe CPU 2
$\overline{\text{WS}}$	2		Write Select, Auswahl "Schreiben" bei DMA-Betrieb
$\overline{\text{MAD}}$ 0...7	1		Memory Address, siehe CPU 1
$\overline{\text{MAD}}$ 8...9	4		
$\overline{\text{MAD}}$ 10...13	3		
MAD		6	Memory Address, RAM/ROM-Baugruppe
MDB		7	Memory Data Bus, siehe CPU 1
DB	1		Data Bus, siehe CPU 1

Tafel A 2.0.4. Signale auf Karte IN/OUT

Signal	TTL-Standard-Lastfaktor F_I	F_O	Erläuterung
IN	1		Input, peripherer Eingang des µR
OUT		10	Output, peripherer Ausgang des µR
$\overline{\text{PORT SEL}}$	2	8	Port Select, Portauswahl
OCS		10	Out-Card Select, Kartenauswahl, dient der Kartenspezifikation
INB		10	Input Bus, siehe CPU 1
MAD' 0...7	2		Memory Address, siehe CPU 1
$\overline{\text{MAD'}}$ 0...7	4	6	Memory Address, IN/OUT-Baugruppe
MAD' 9...10	4		Memory Address, siehe CPU 1
MAD' 11	2		
MAD' 12	4		
MAD'' 11	9	12	Memory Address, IN/OUT-Baugruppe
$\overline{\text{I/O OUT}}$	1		IN/OUT Output, siehe CPU 2
I/O OUT		9	IN/OUT Output, IN/OUT-Baugruppe

Tafel A 2.0.5. Signale auf Karte OUT

Signal	TTL-Standard-Lastfaktor F_I	F_O	Erläuterung
OUT		10	Output, peripherer Ausgang des µR
MAD'	2		Memory Address, siehe CPU 1
CARD SEL 1/2	2		Card Select, Kartenauswahl
$\overline{\text{I/O OUT}}$	1		IN/OUT Output, siehe CPU 2
$\overline{\text{PORT SEL}}$	2		Port Select, Portauswahl

Tafel A 2.0.6. Signale auf Karte INT

Signal	TTL-Standard-Lastfaktor		Erläuterung
	F_I	F_O	
IA	1		Interruptanmeldung
INT INH	1		Interrupt Inhibit, Interrupthemmung
$\overline{\text{INT REQ}}$		10	Interrupt Request, Interruptanforderung, siehe CPU 2
$\overline{\text{T2}}$, $\overline{\text{T3}}$	1		Statussignale, siehe CPU 2
$\overline{\text{T3A}}$	1		siehe CPU 2
T3A		9	
IIB		10	Interrupt Input Bus, siehe CPU 1
$\overline{\text{IEND}}$	1		Interrupt End, Interruptende-Anweisung (über OUT-Tor)
CLK	1		Clock, Takteingang der Interruptbaugruppe
HIGH		25	High, H-Potential
$\overline{\text{INT SEL}}$	1		Interrupt Select, siehe CPU 2
$\overline{\text{INT DEL}}$		10	Interrupt Delay, verzögerte Interruptende-Information
$\overline{\text{CLA}}$	1		Clear All, siehe CPU 2

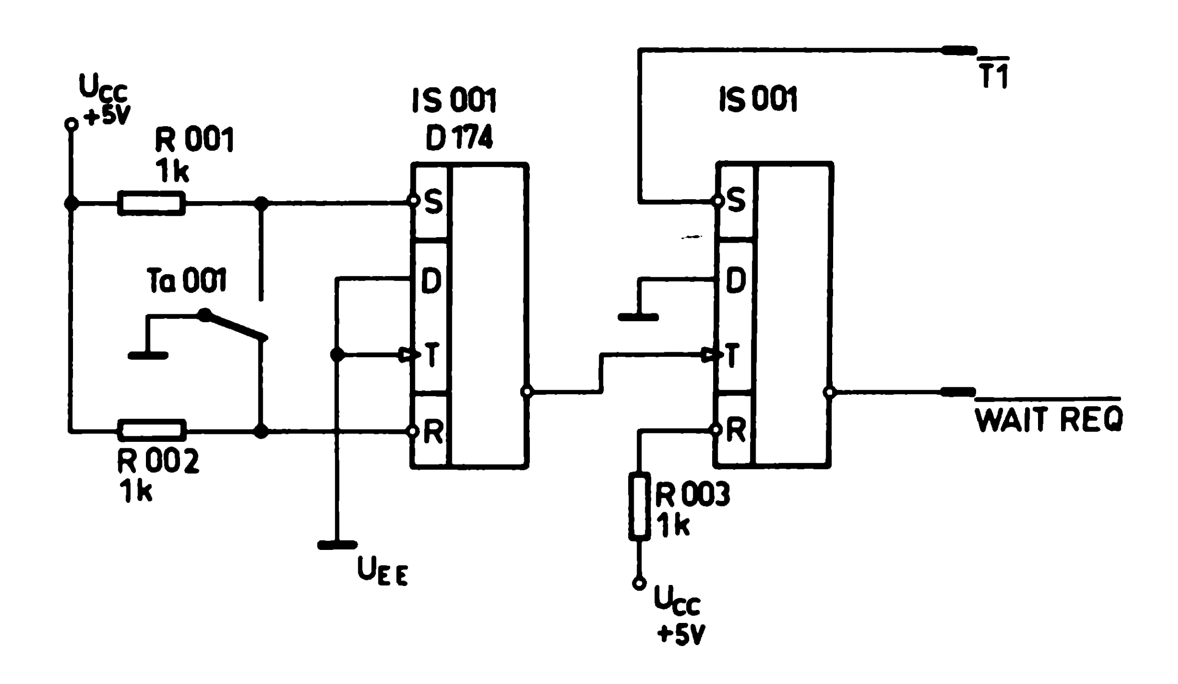


Bild A 2.1. SST - Baugruppe

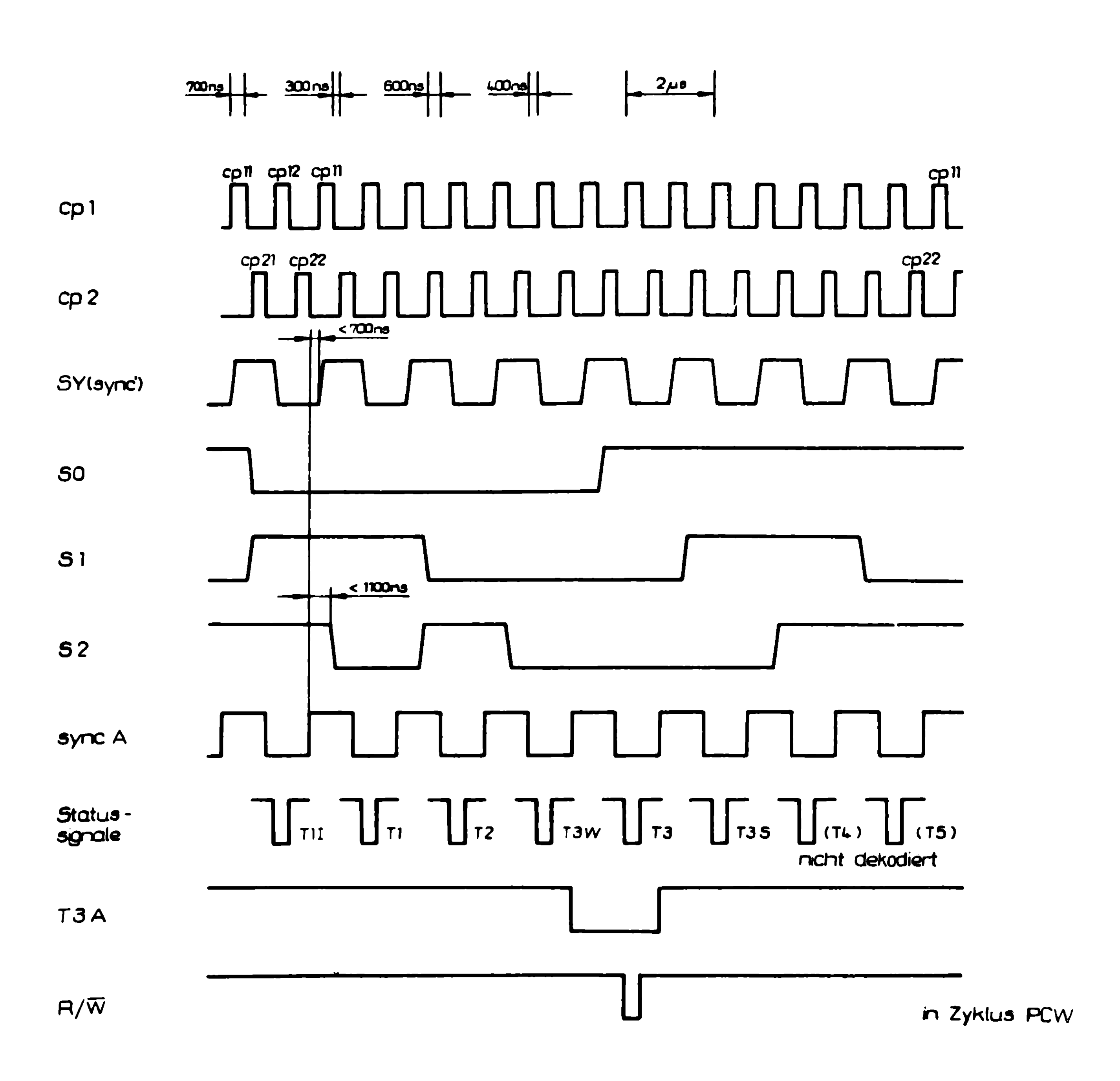


Bild A2.2. Impulsdiagramm der CPU-Baugruppe

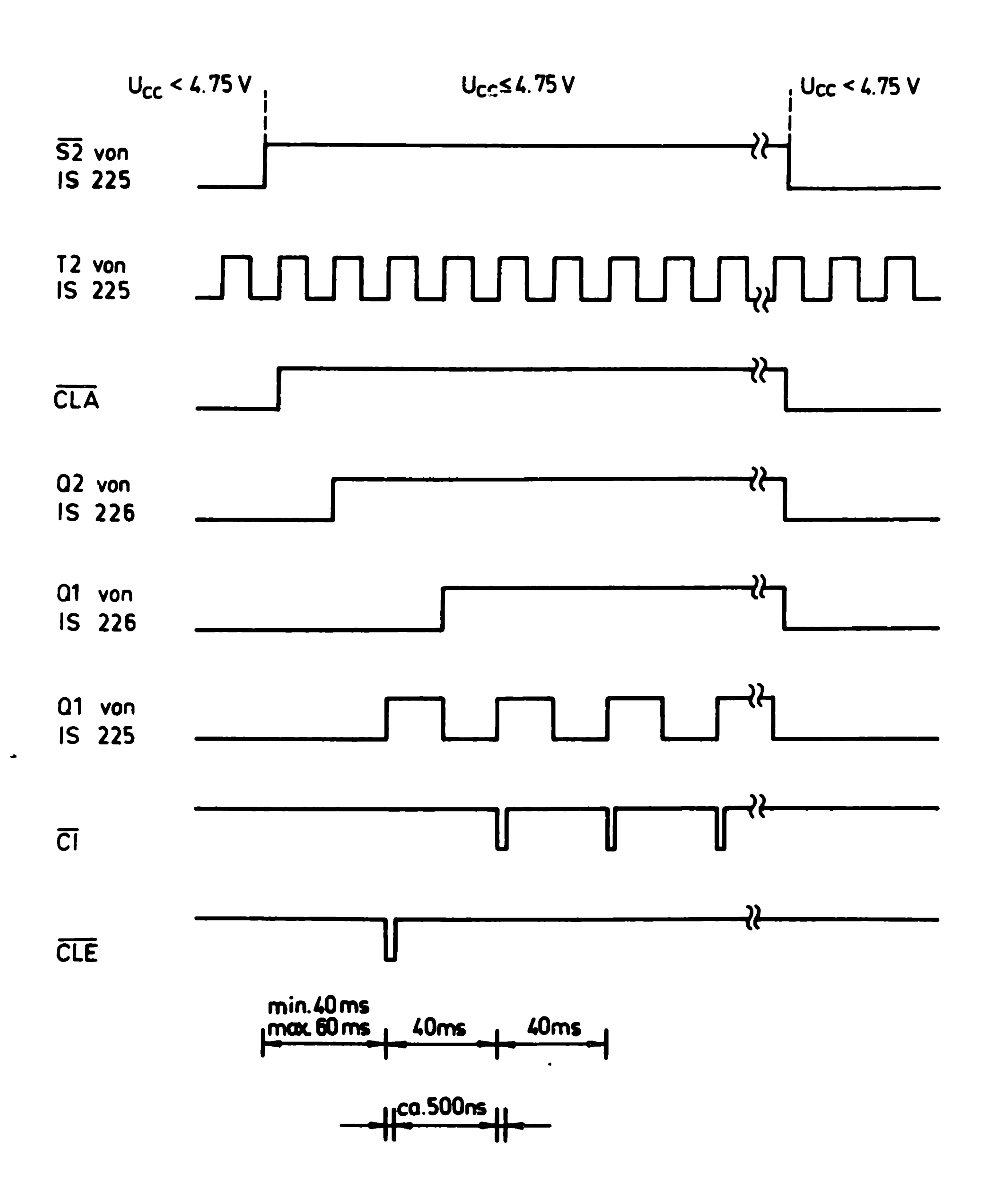


Bild A 2. 3. Impulsdiagramm der Anlauflogik

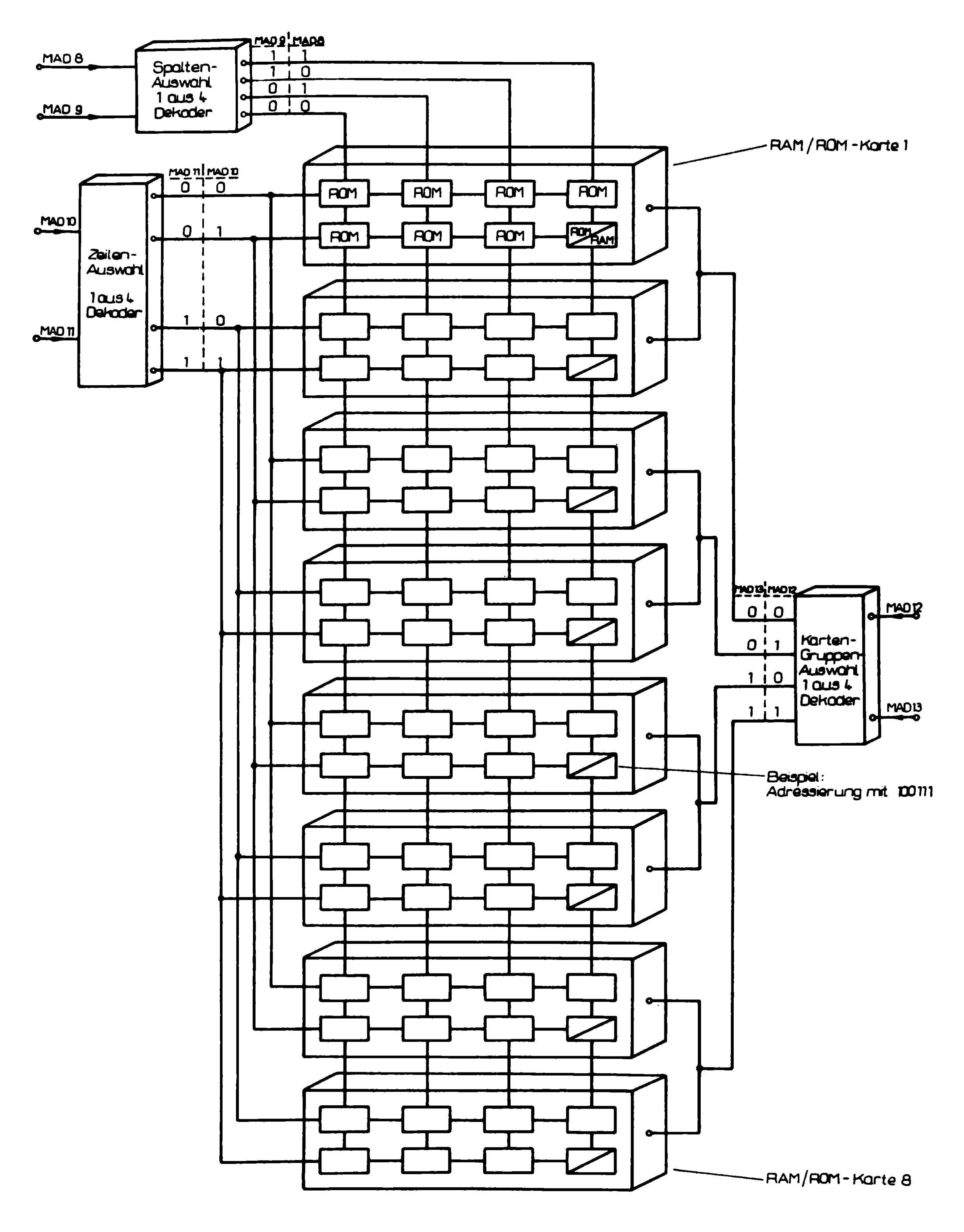


Bild A2.4 Organisation der Speicherbaugruppe

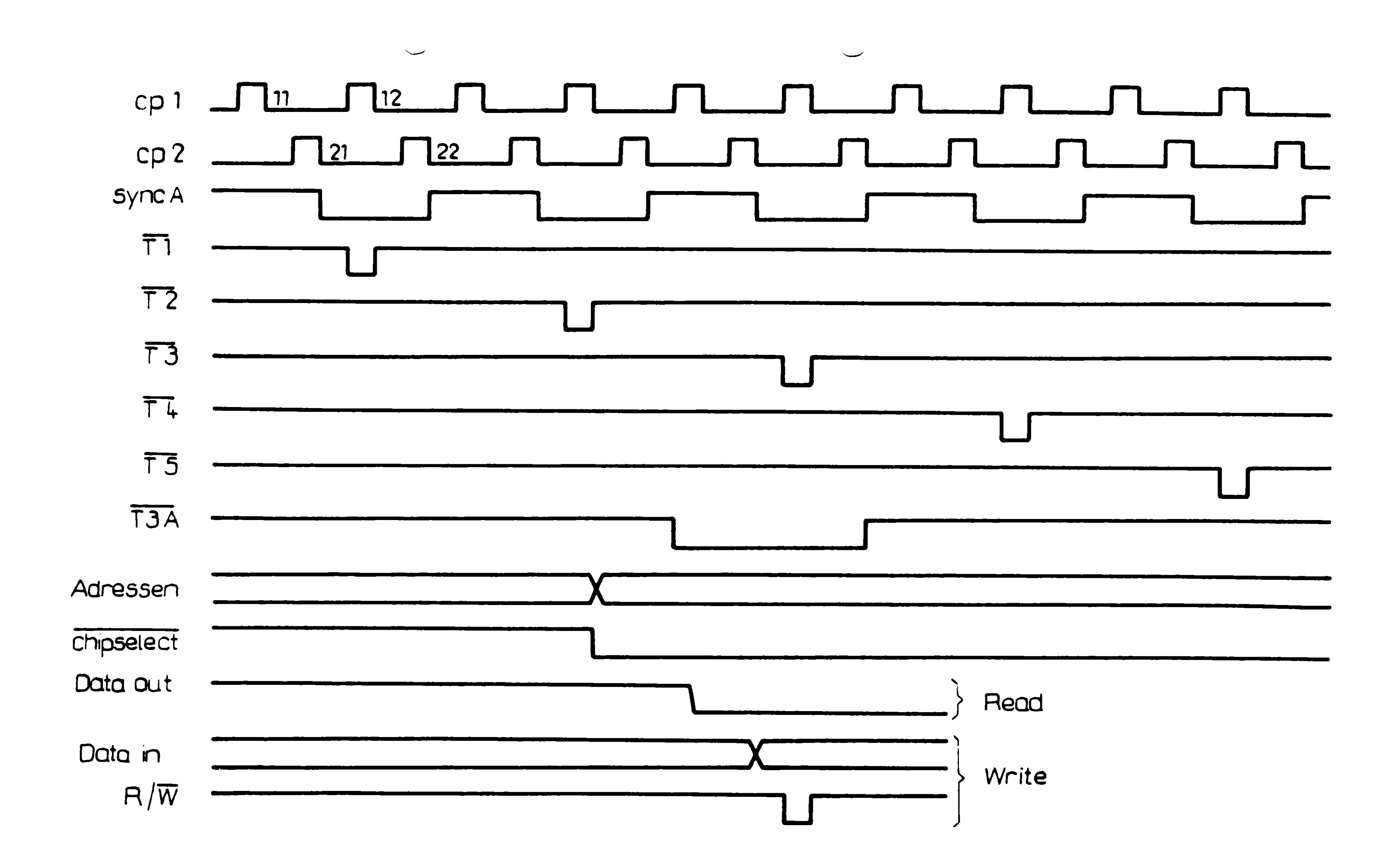


Bild A 2.5. **Impulsdiagramm der Speicherbaugruppe**

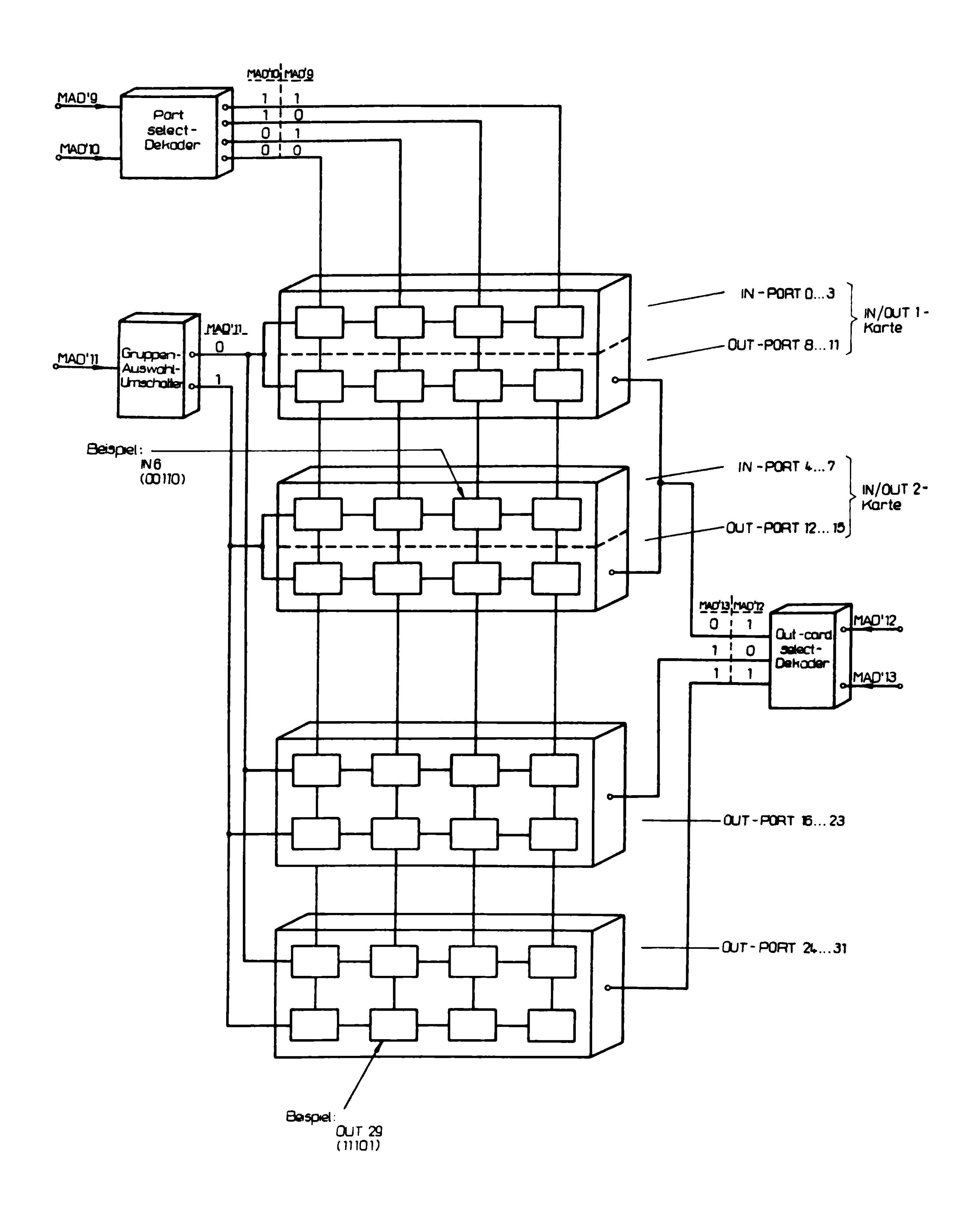


Bild A 2.5. Organisation der Ein-/Ausgabebaugruppe

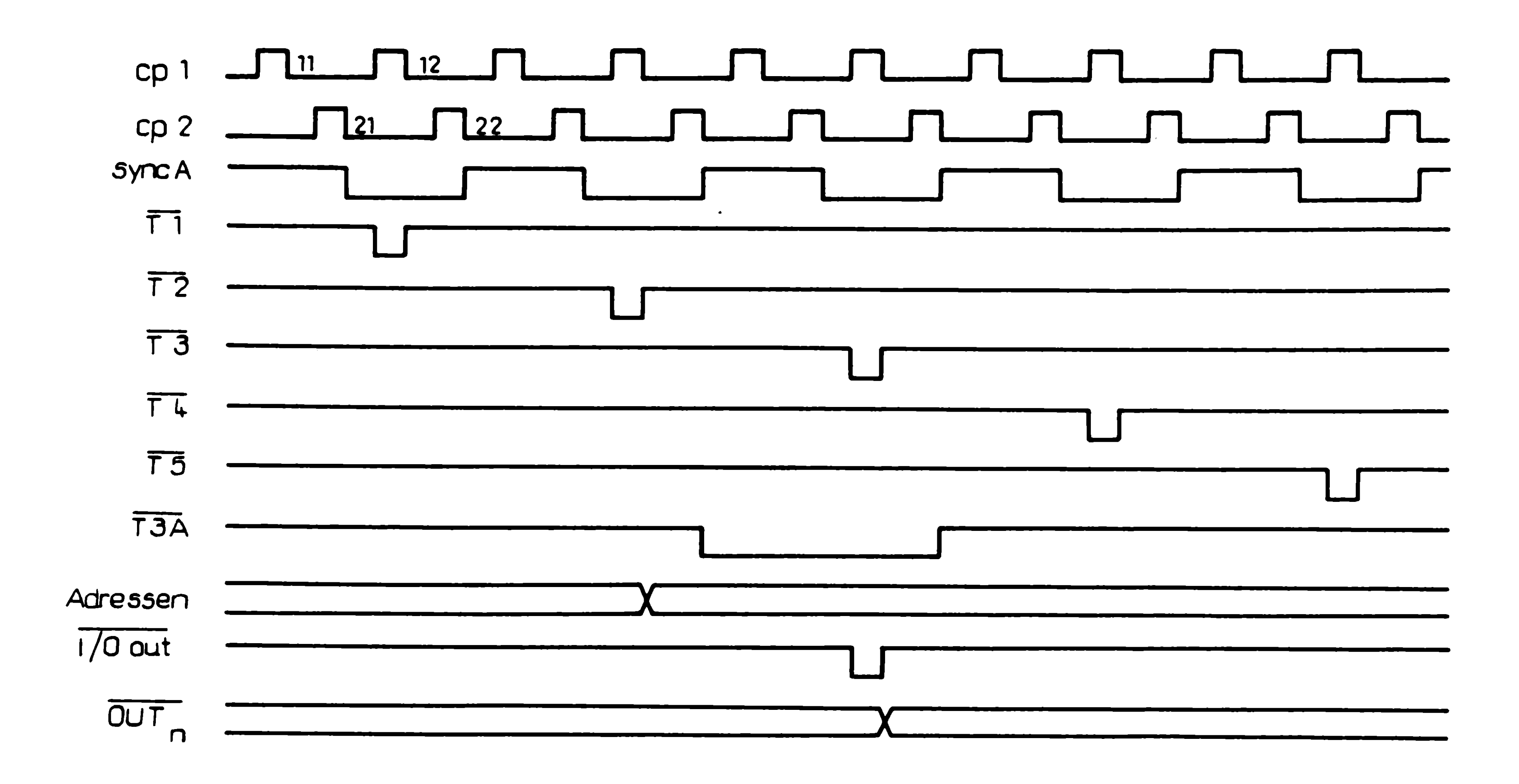


Bild A2.7. Impulsdiagramm der Ein-/Ausgabebaugruppe

A 2.8. Impulsdiagramme Interruptbehandlung

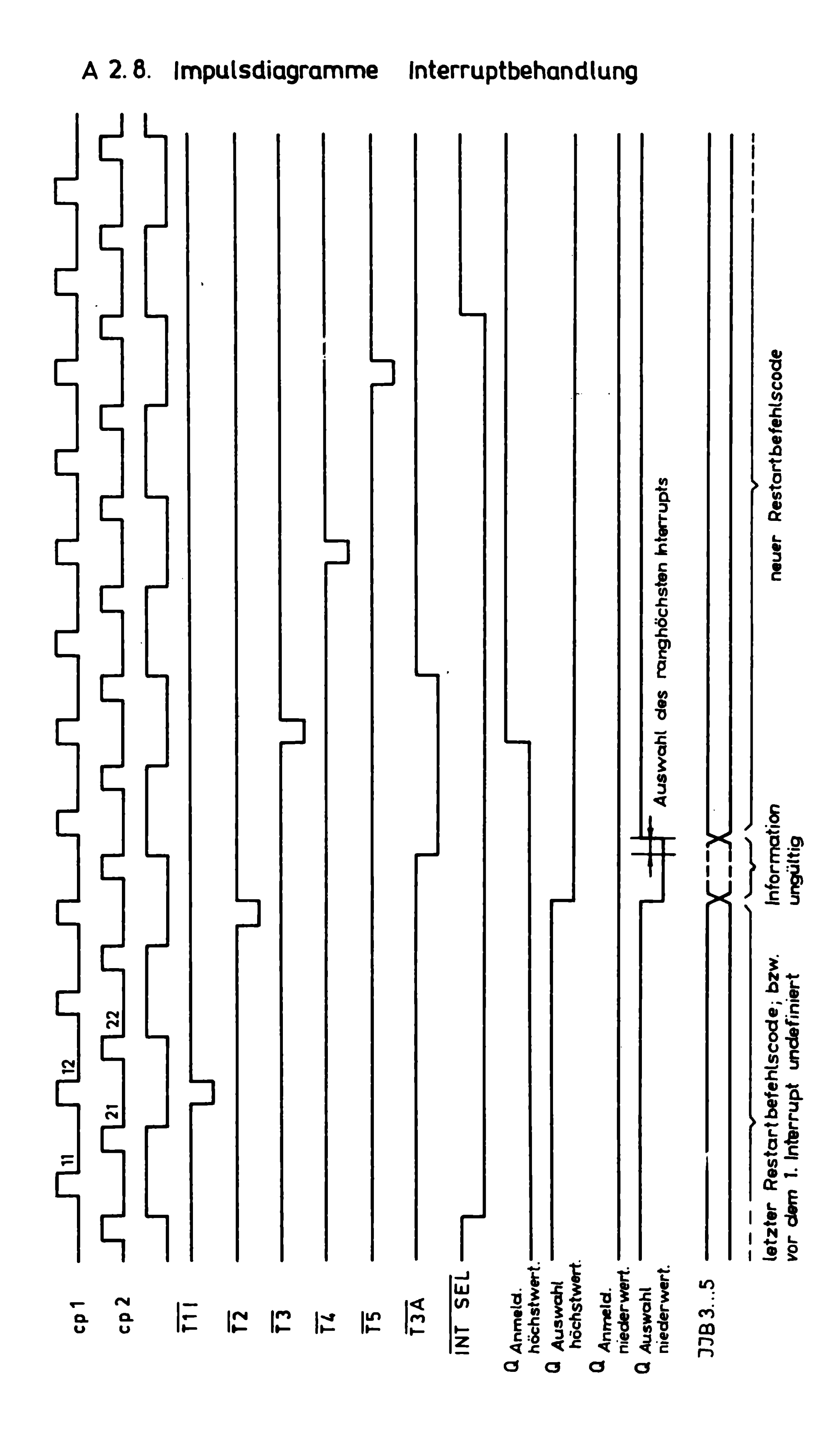


Bild A 2.8.1. Impulsdiagramm der Interruptbaugruppe

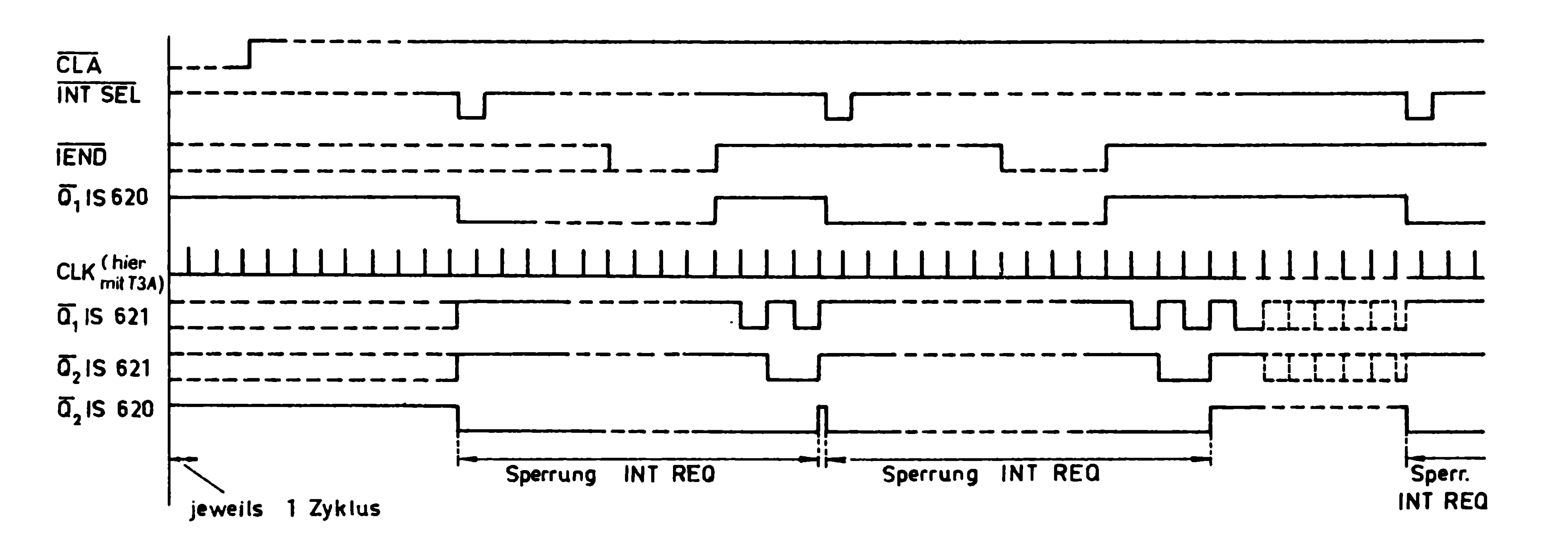
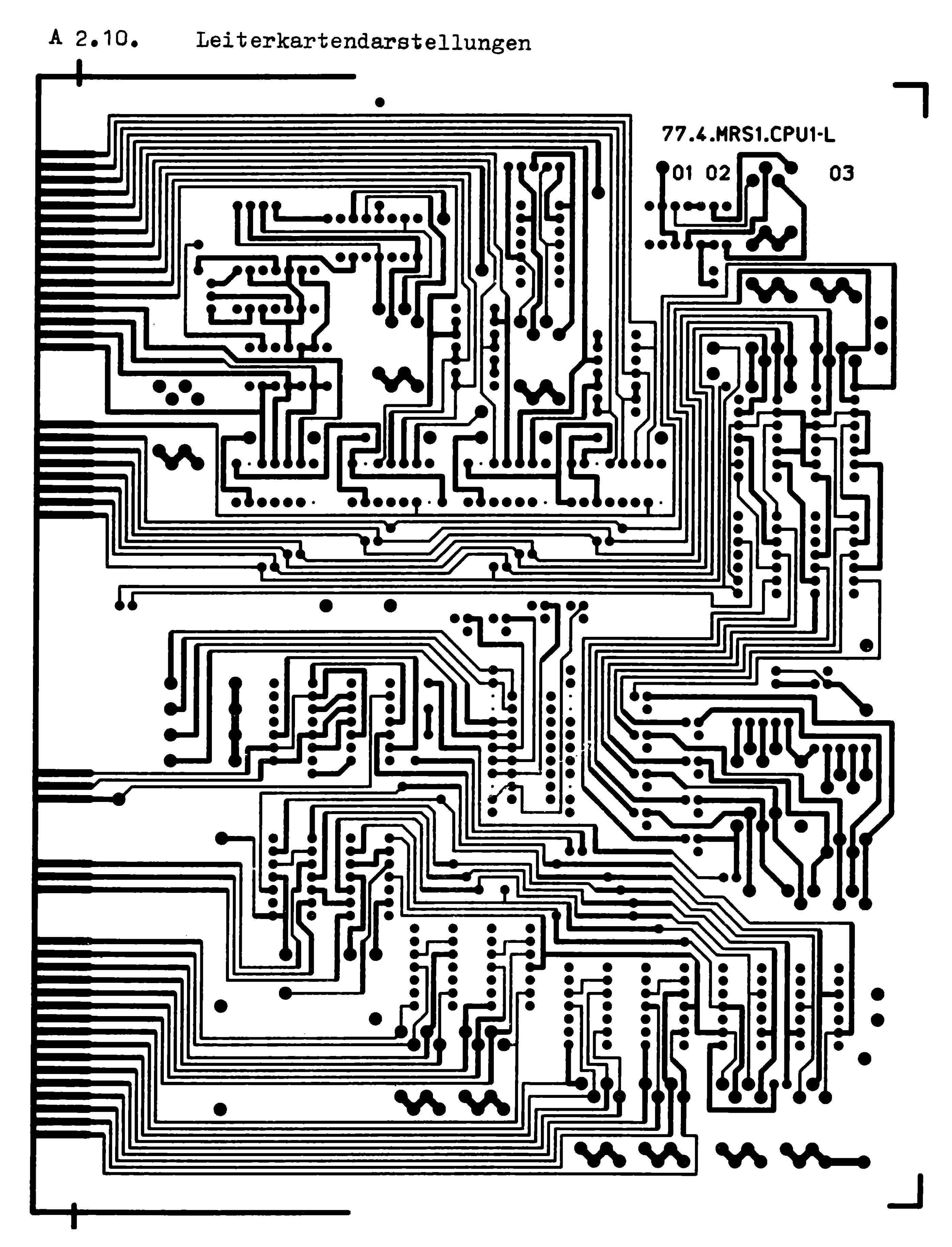
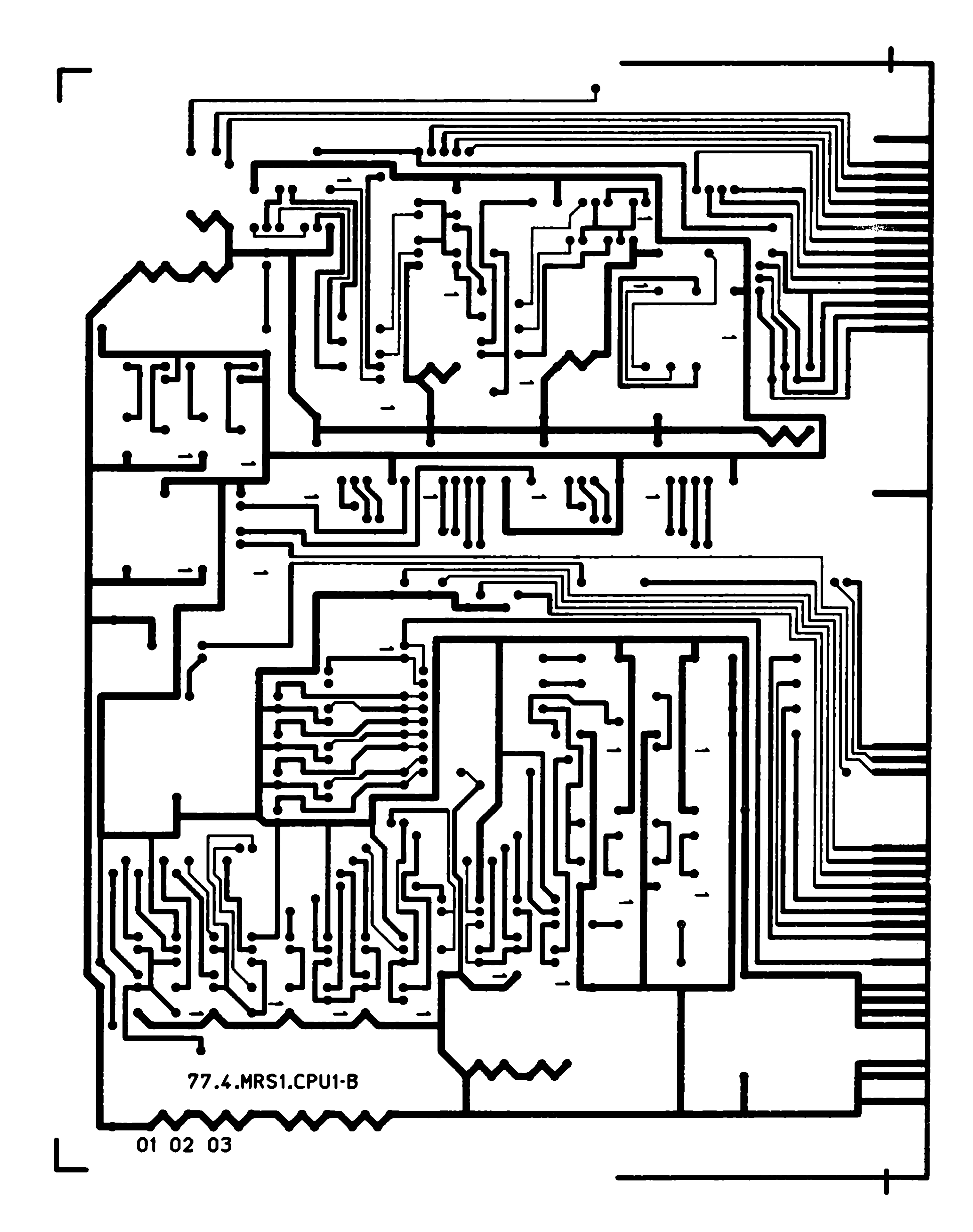


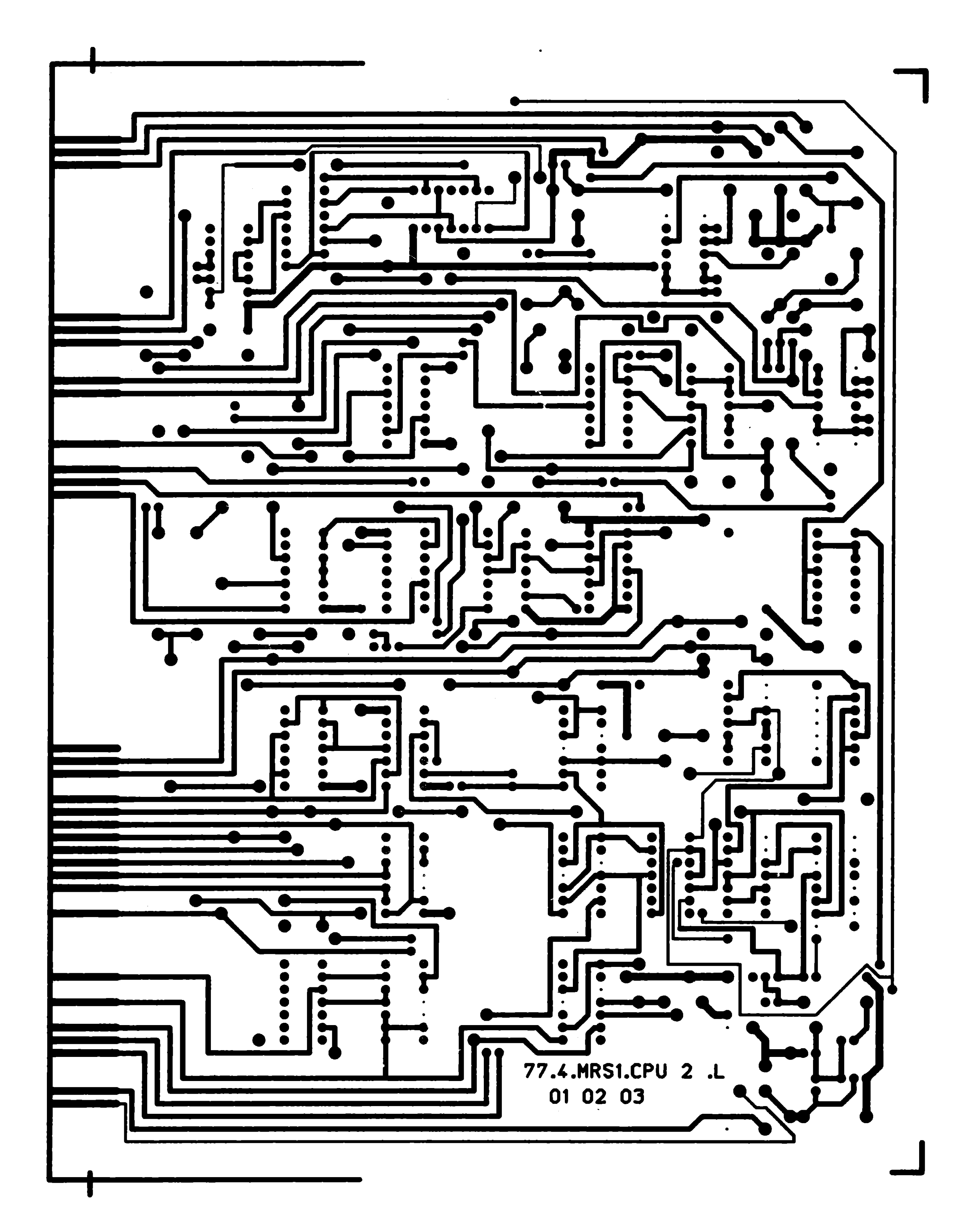
Bild A 2.8.2. Impulsdiagramm der ISR



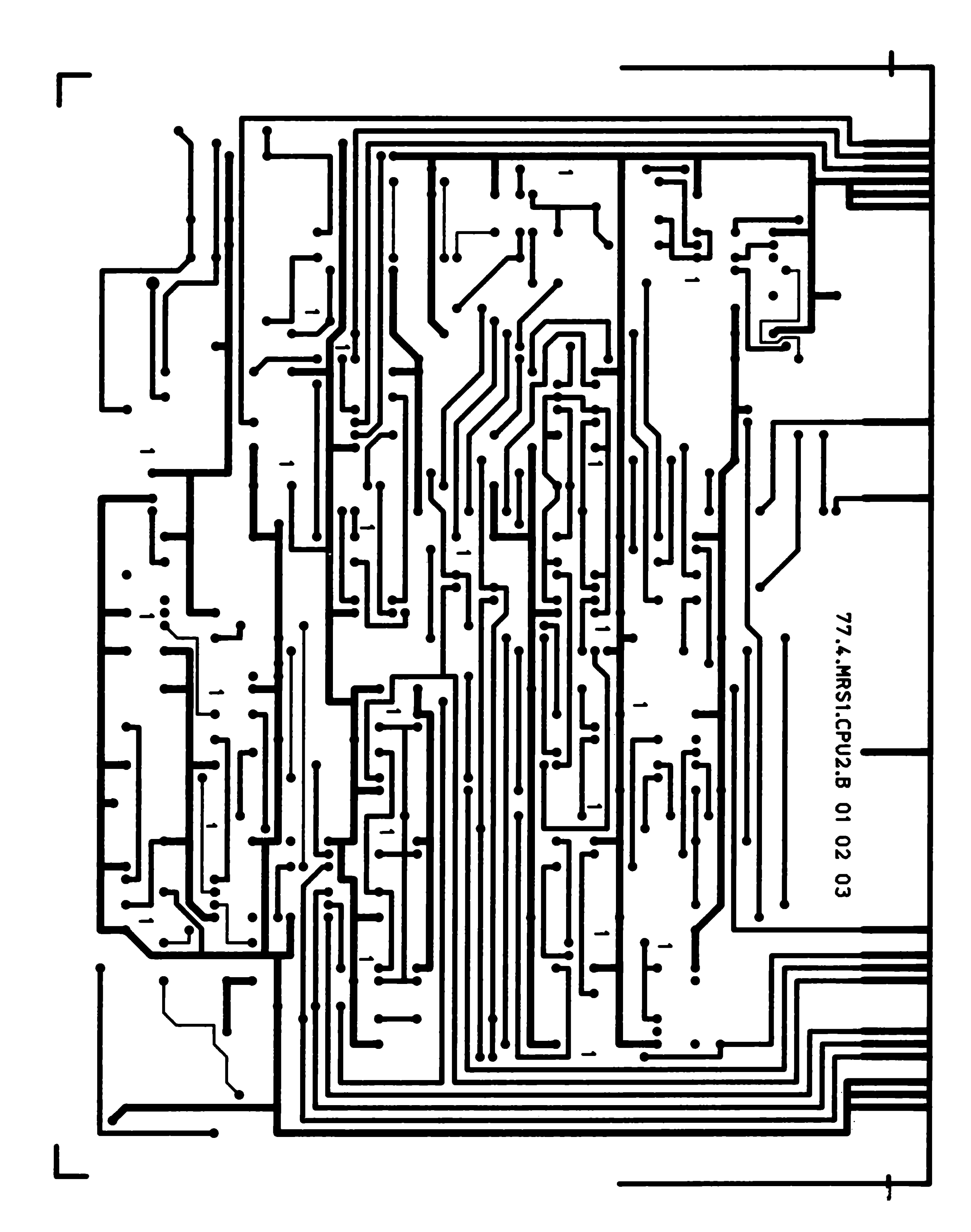
A 2.10.1. Leiterplatte CPU 1, Lötseite



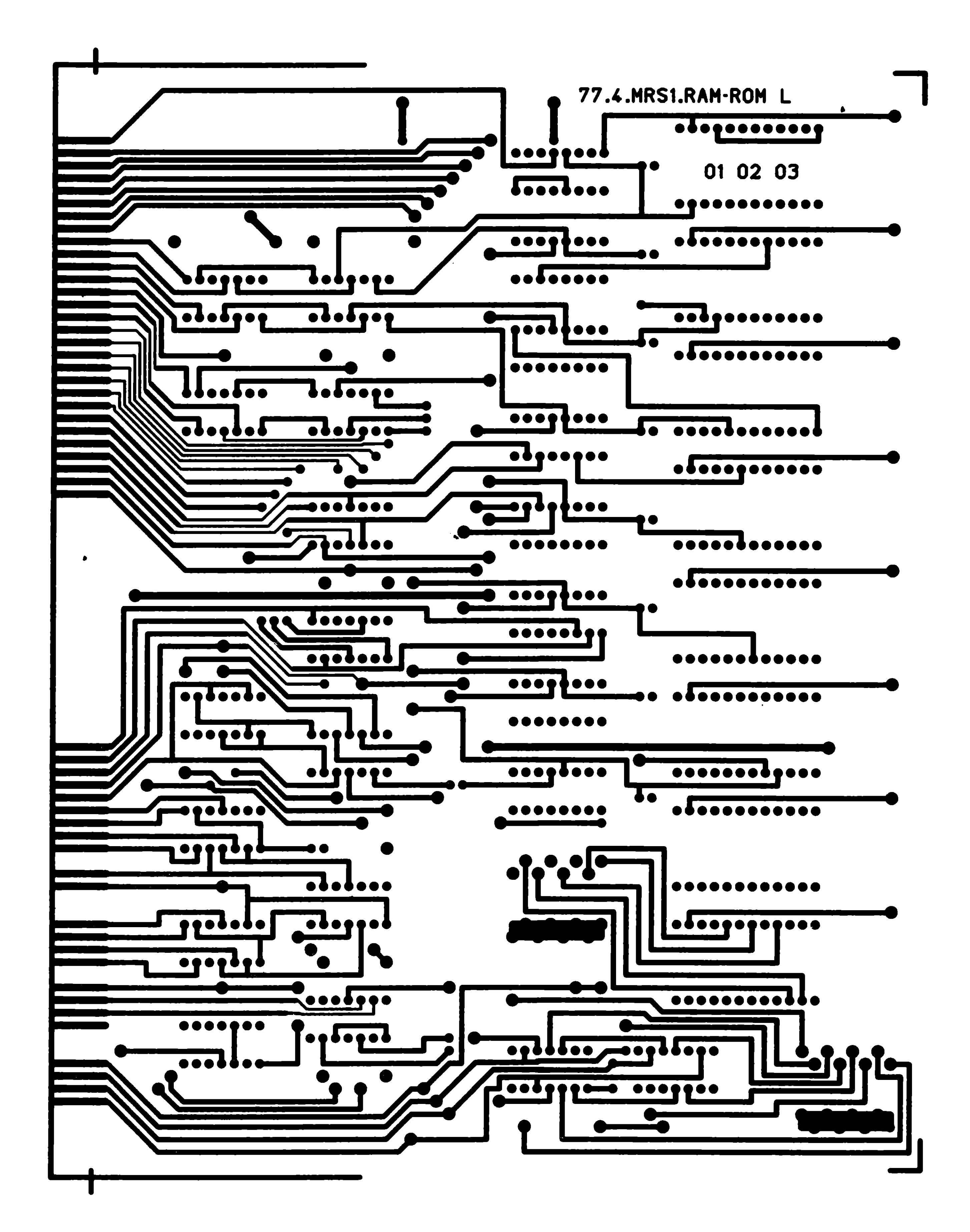
A 2.10.2. Leiterplatte CPU 1, Bestückungsseite



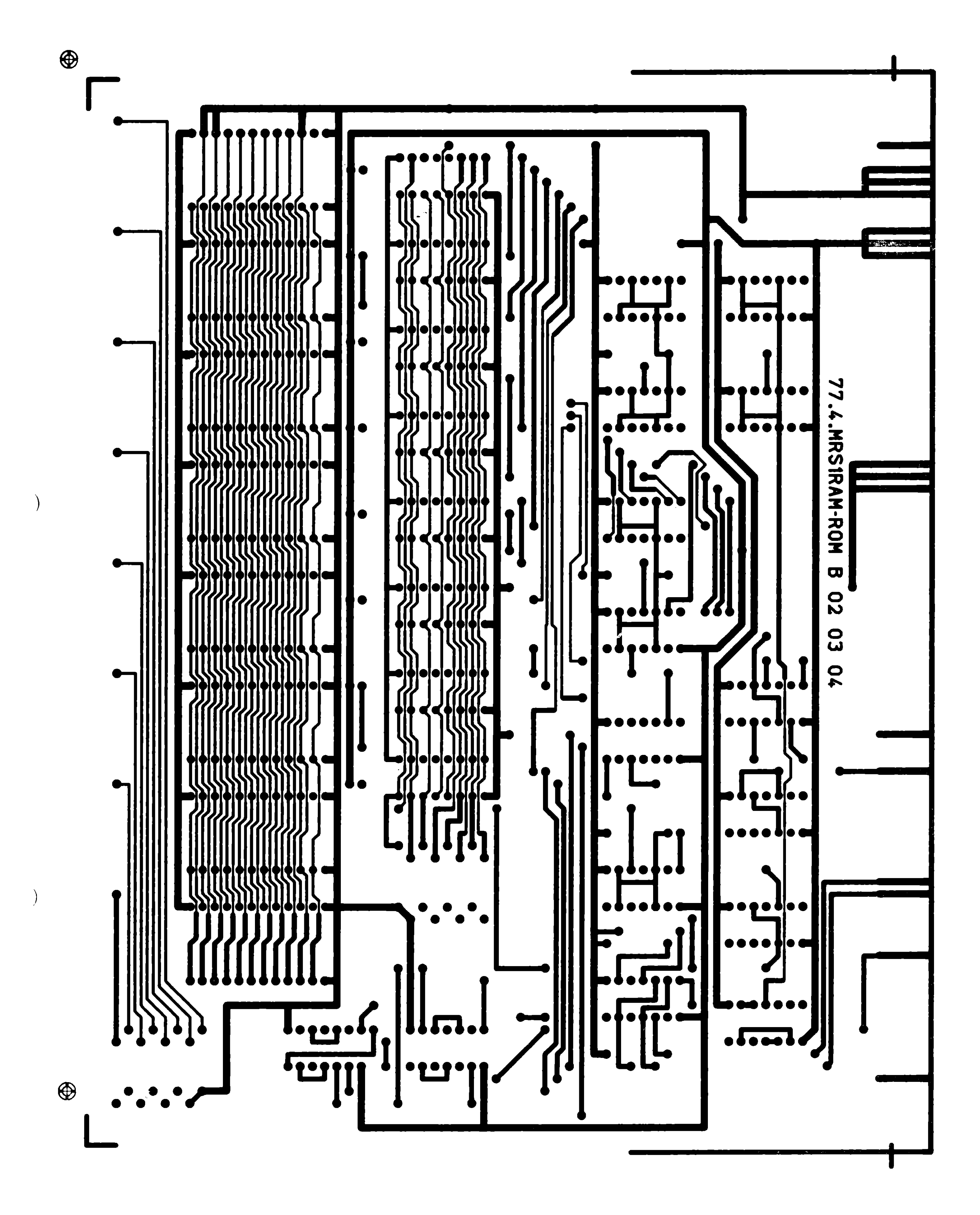
A 2.10.3. Leiterplatte CPU 2, Lötseite



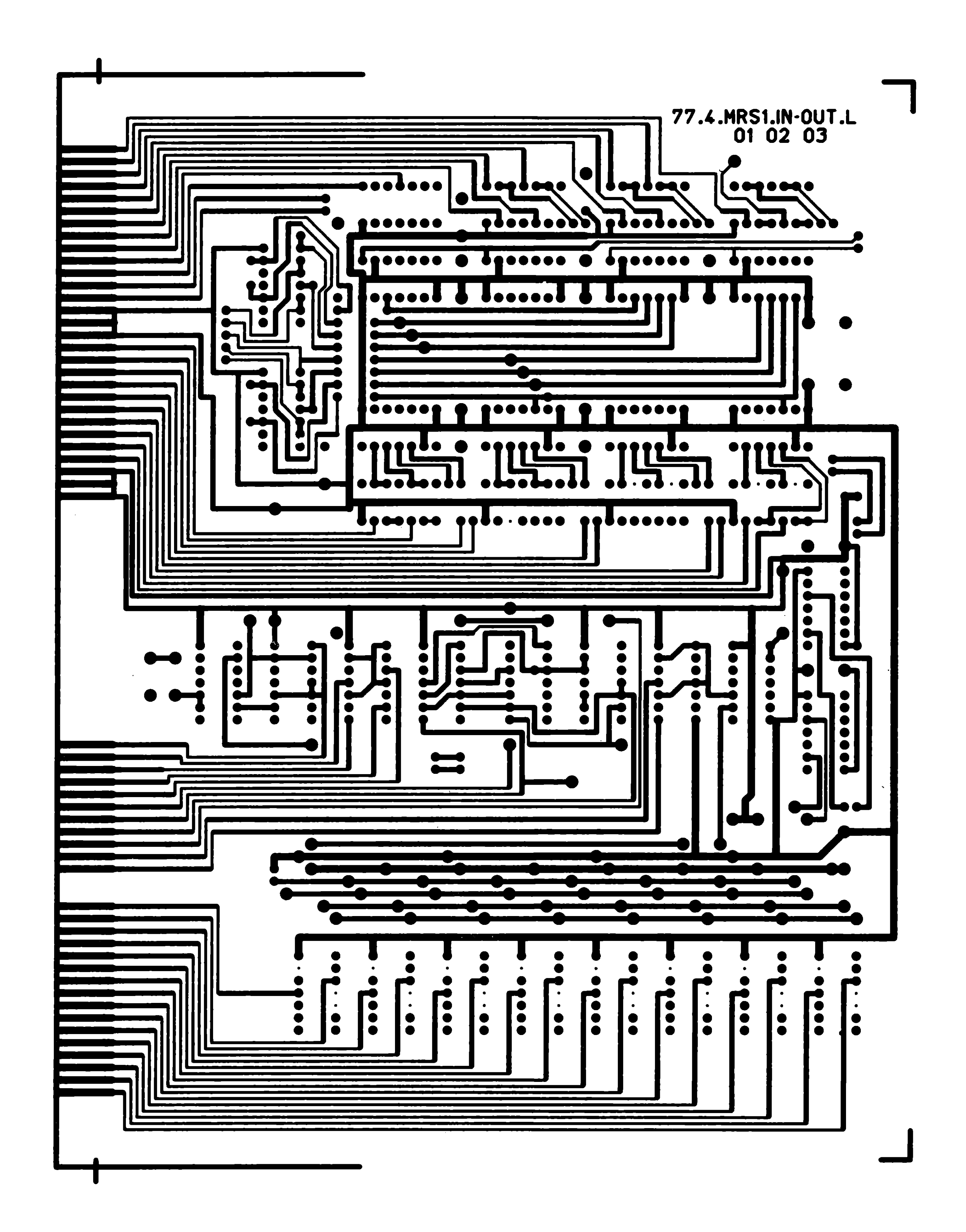
A 2.10.4. Leiterplatte CPU 2, Bestückungsseite



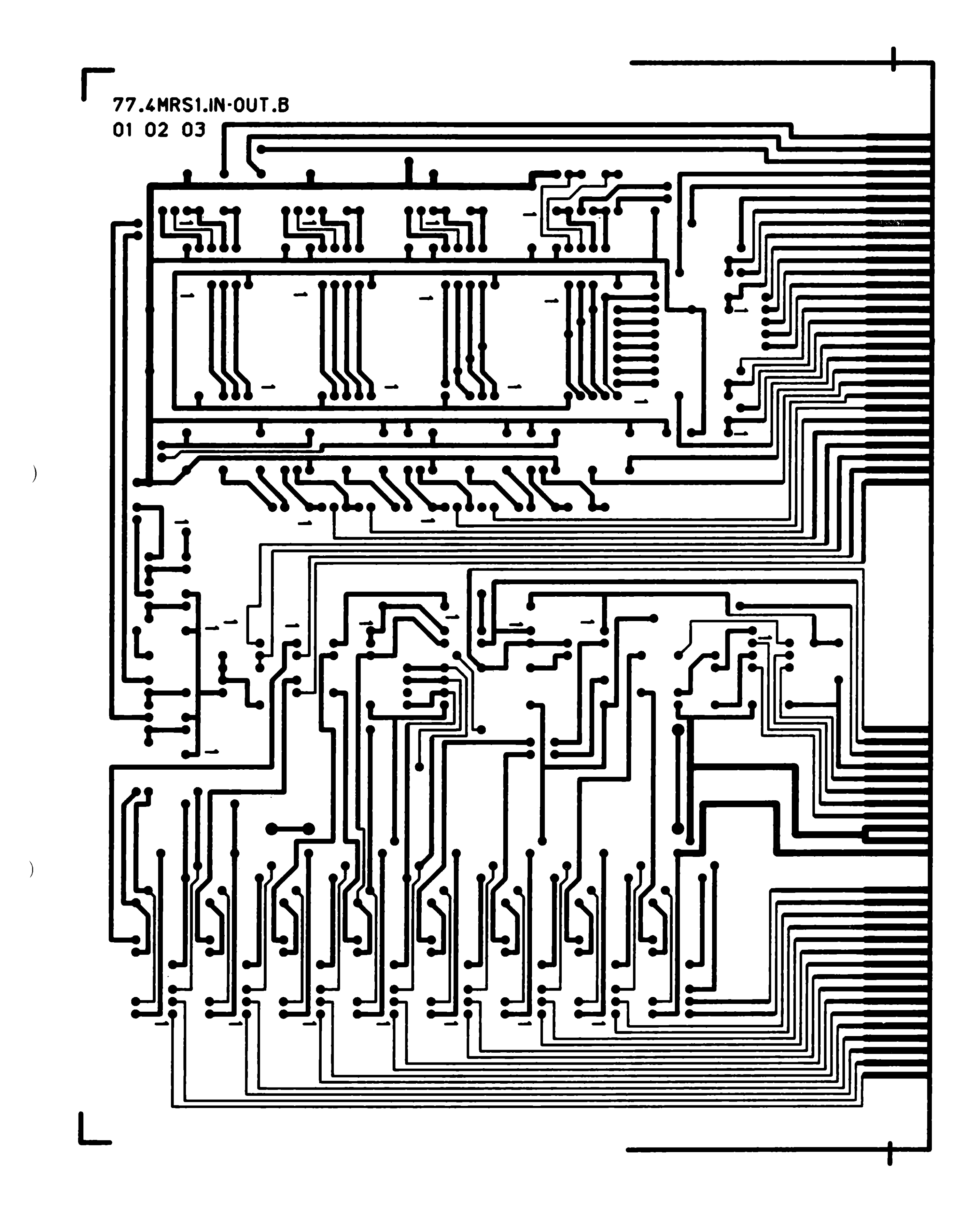
A 2.10.5. Leiterplatte RAM/ROM, Lötseite



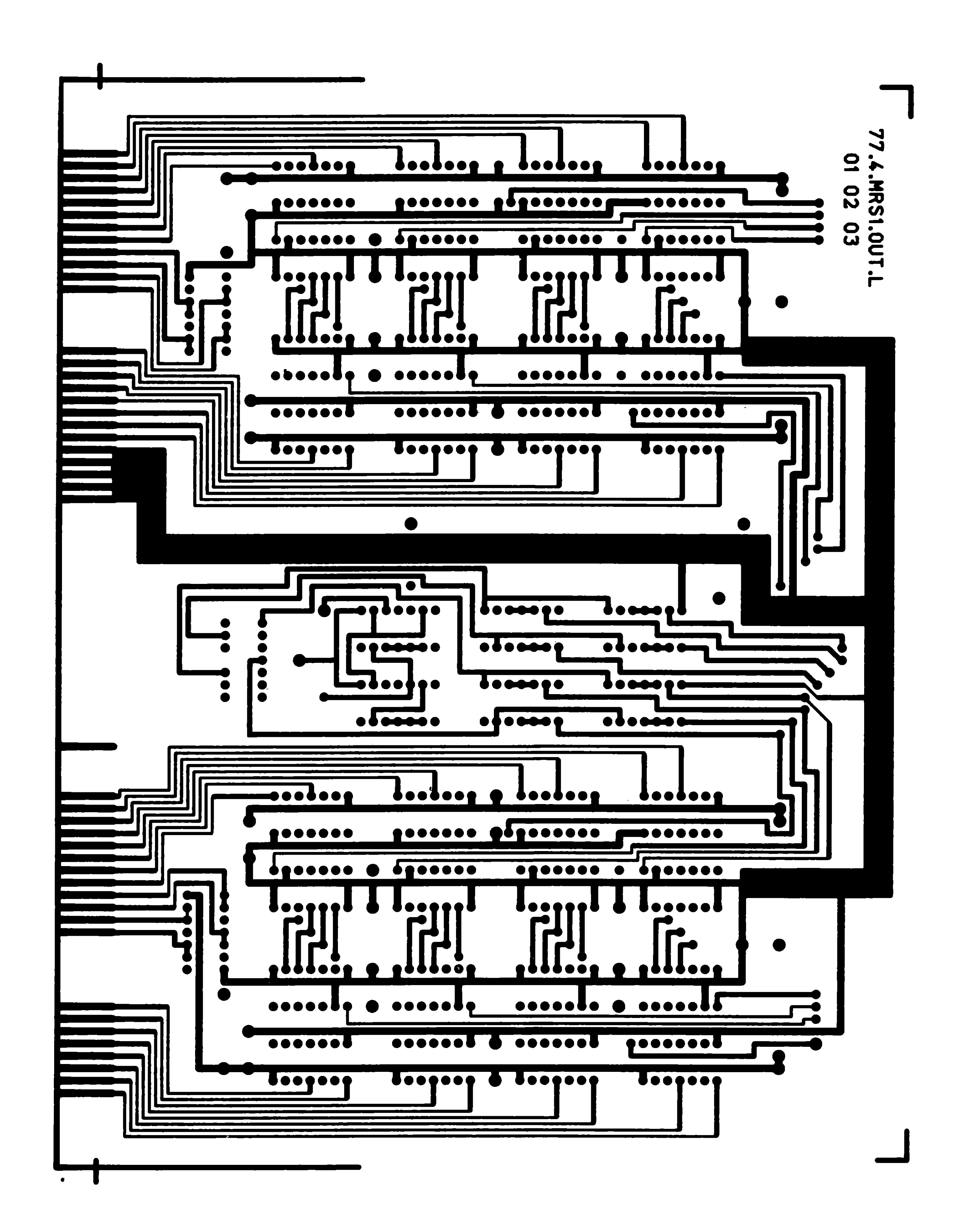
A 2.10.6. Leiterplatte RAM/ROM, Bestückungsseite



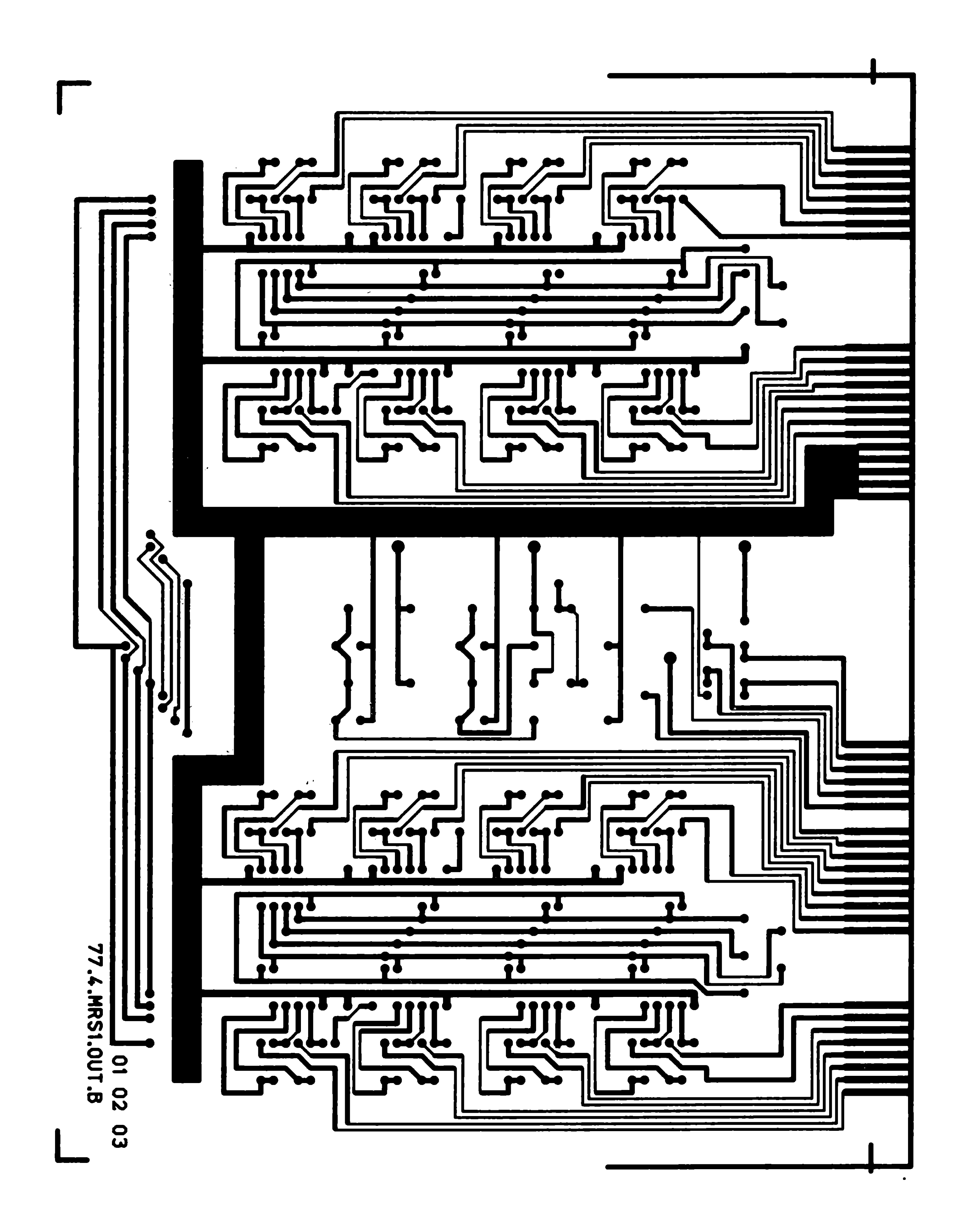
A 2.10.7. Leiterplatte IN/OUT, Lötseite



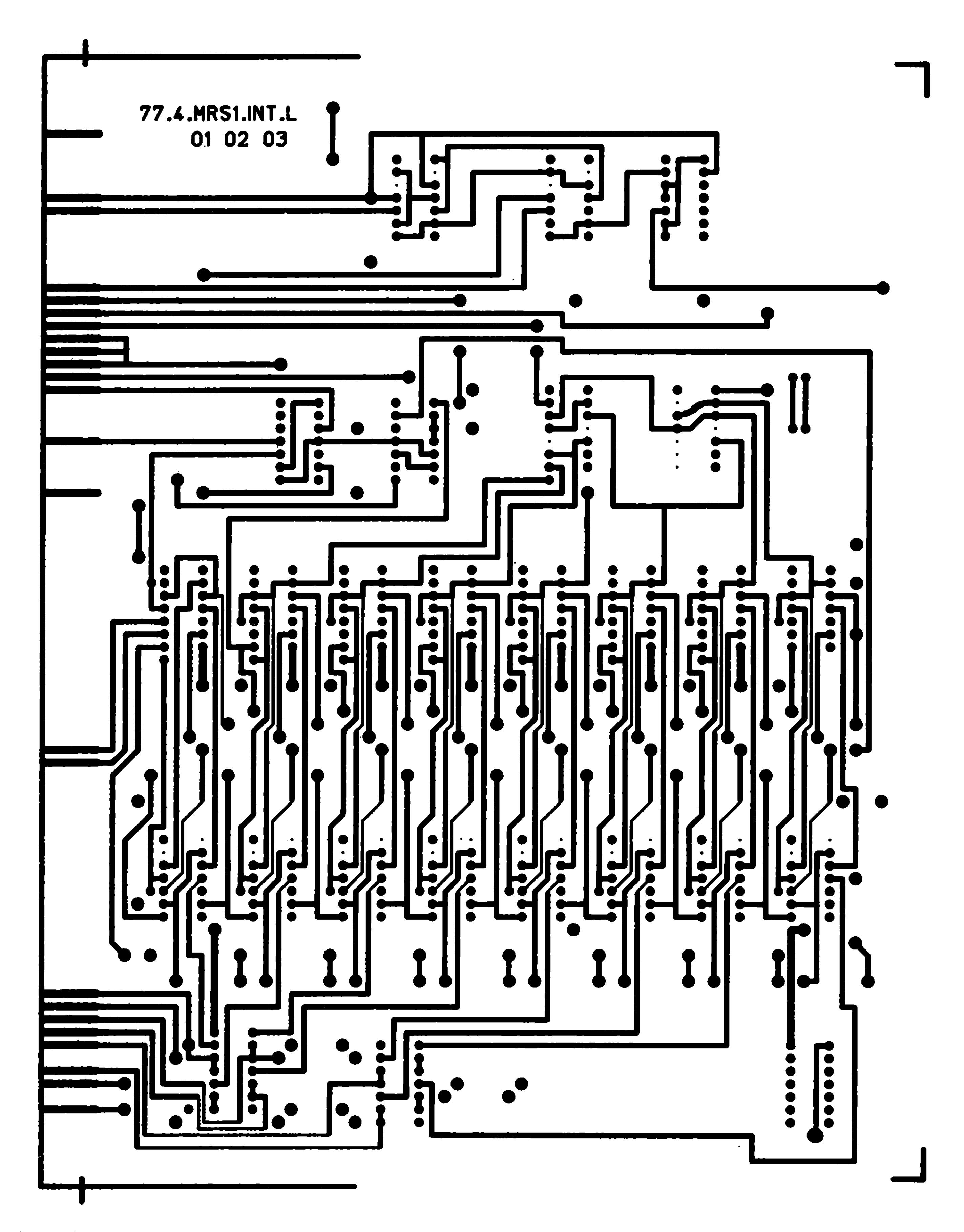
A 2.10.8. Leiterplatte IN/OUT, Bestückungsseite



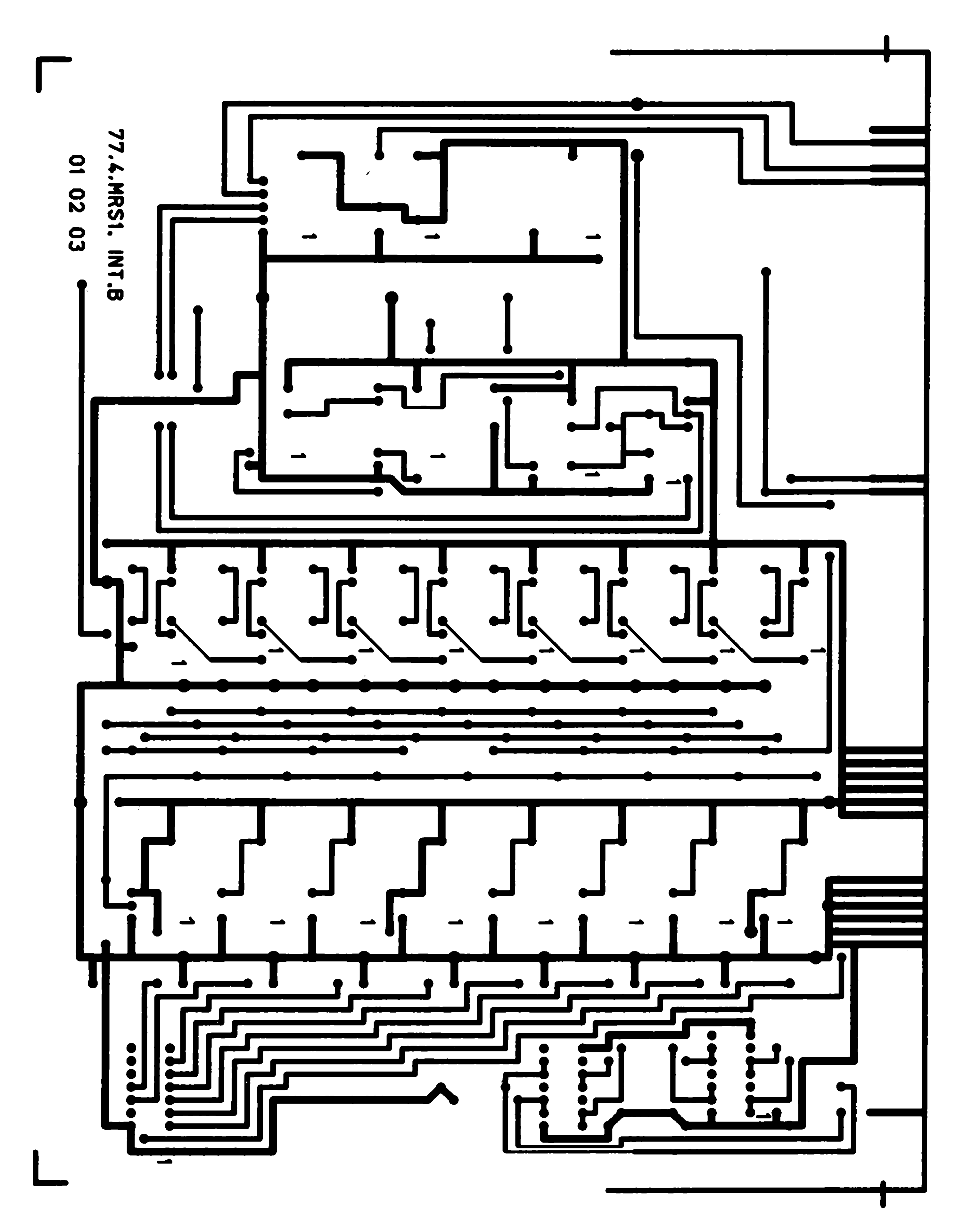
A 2.10.9. Leiterplatte OUT, Lötseite



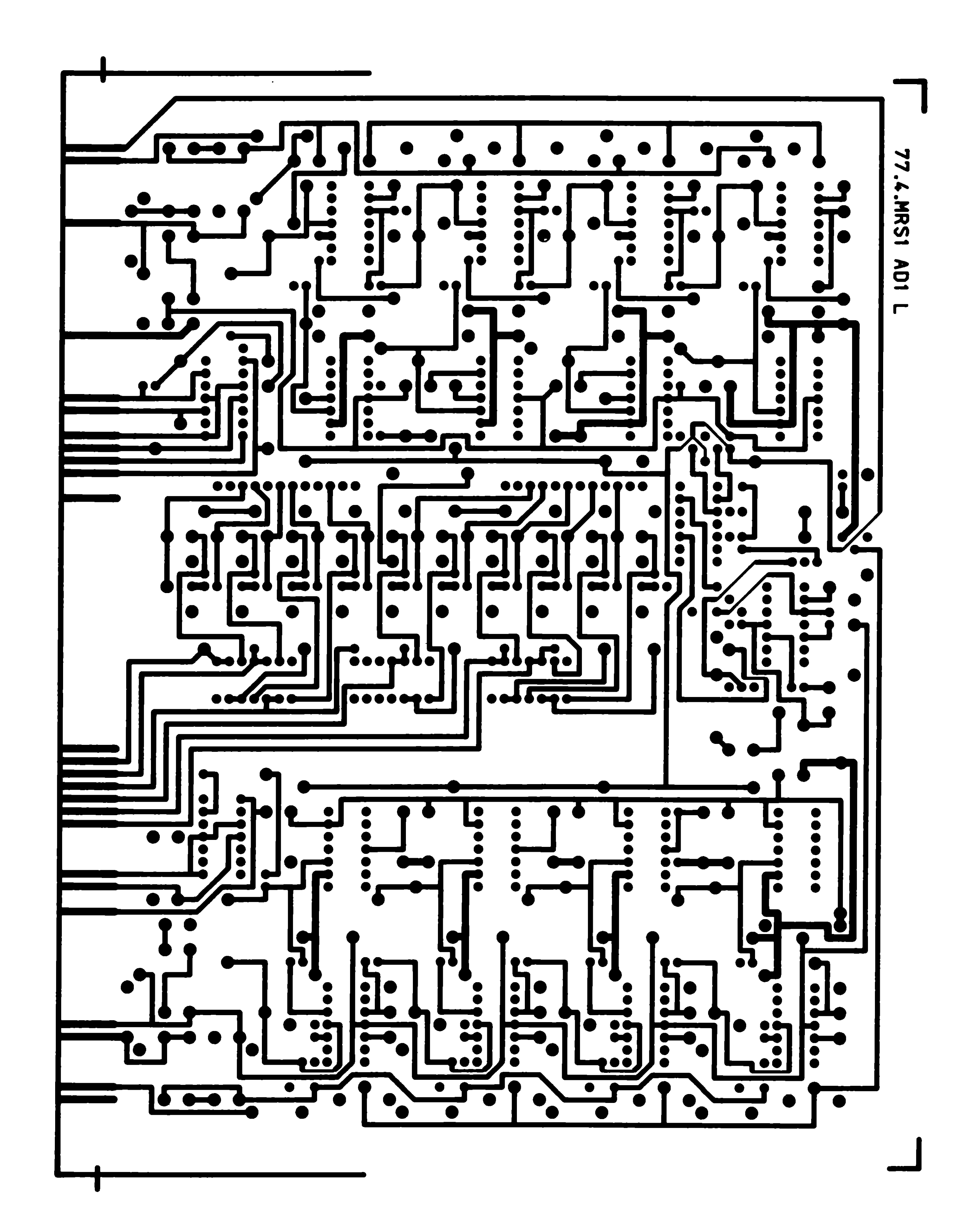
A 2.10.10. Leiterplatte OUT, Bestückungsseite



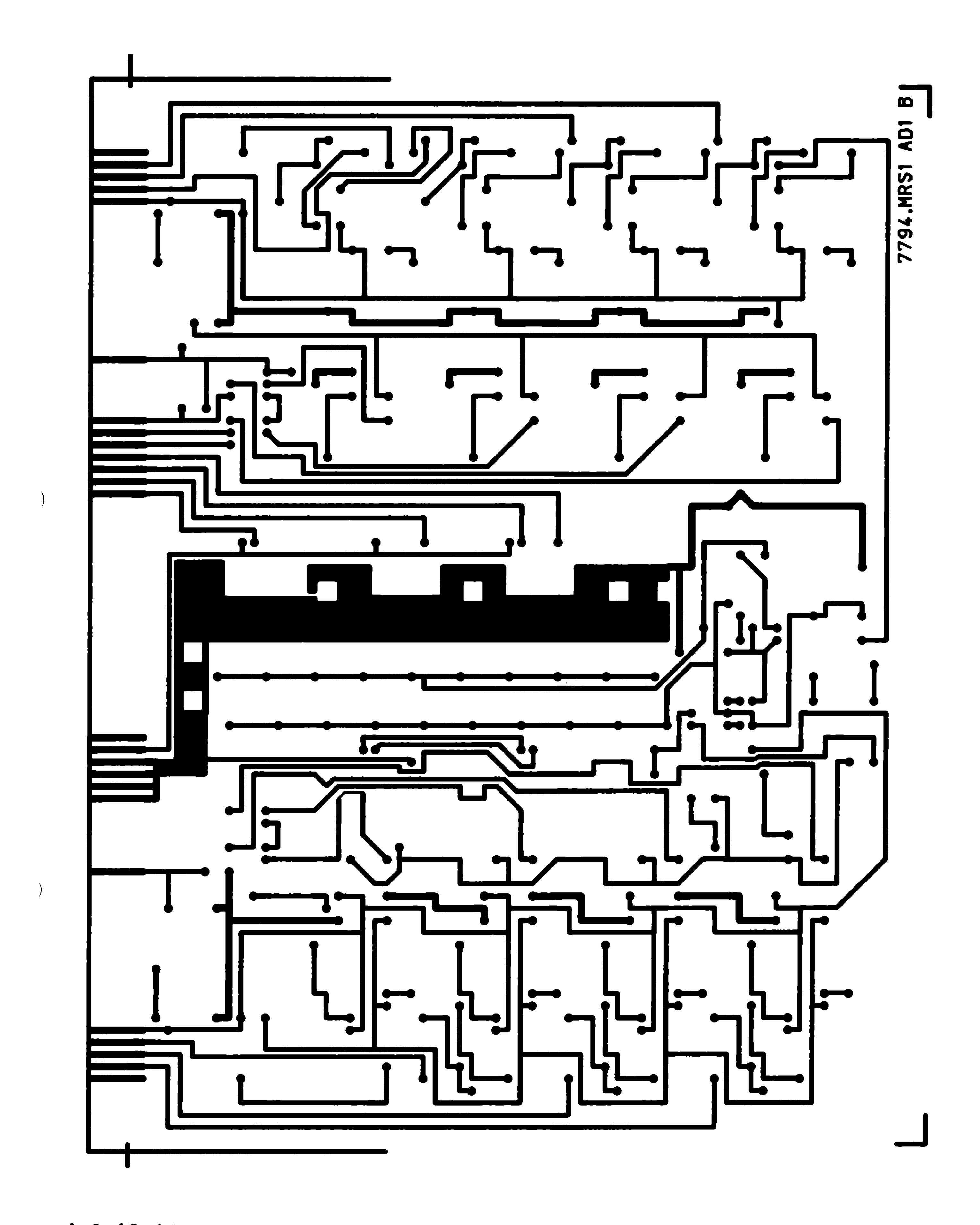
A 2.10.11. Leiterplatte INT, Lötseite



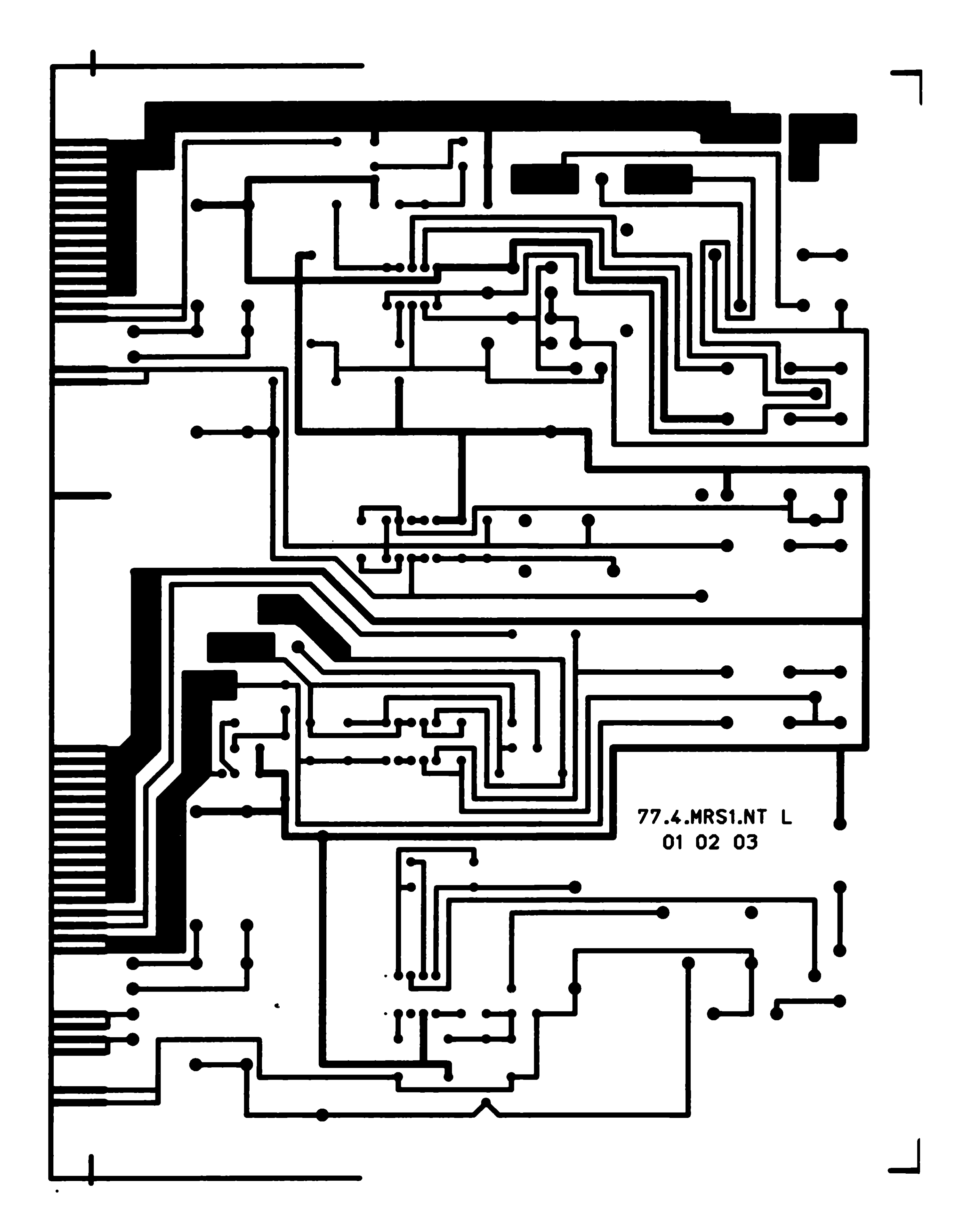
A 2.10.12. Leiterplatte INT, Bestückungsseite



A 2.10.13. Leiterplatte AD, Lötseite



A 2.10.14. Leiterplatte AD, Bestückungsseite



A 2.10.15. Leiterplatte NT, Lötseite

A 2.11. Bestückungspläne der Leiterkarten

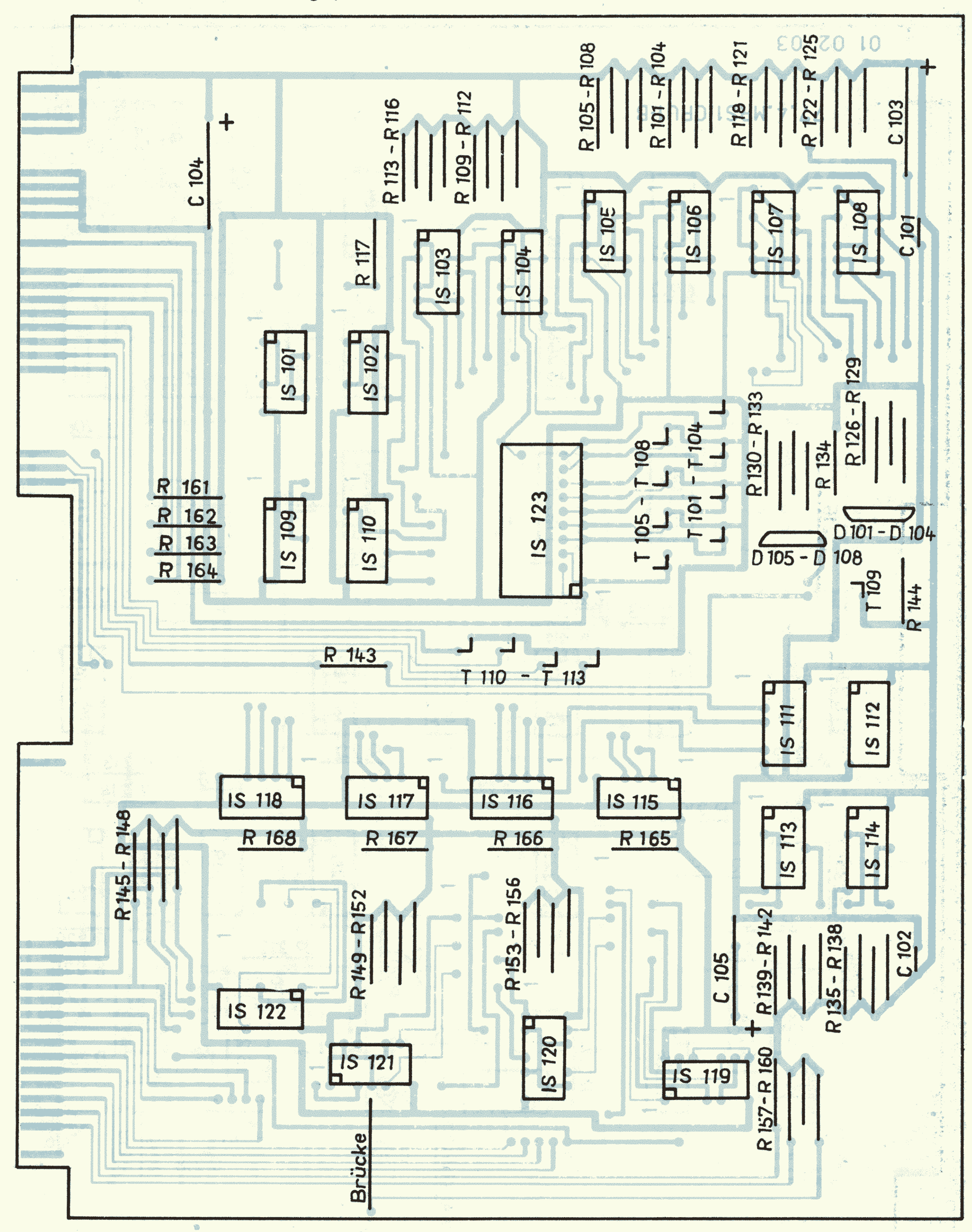


Bild A 2.11.1. Bestückungsplan CPU 1 - Karte

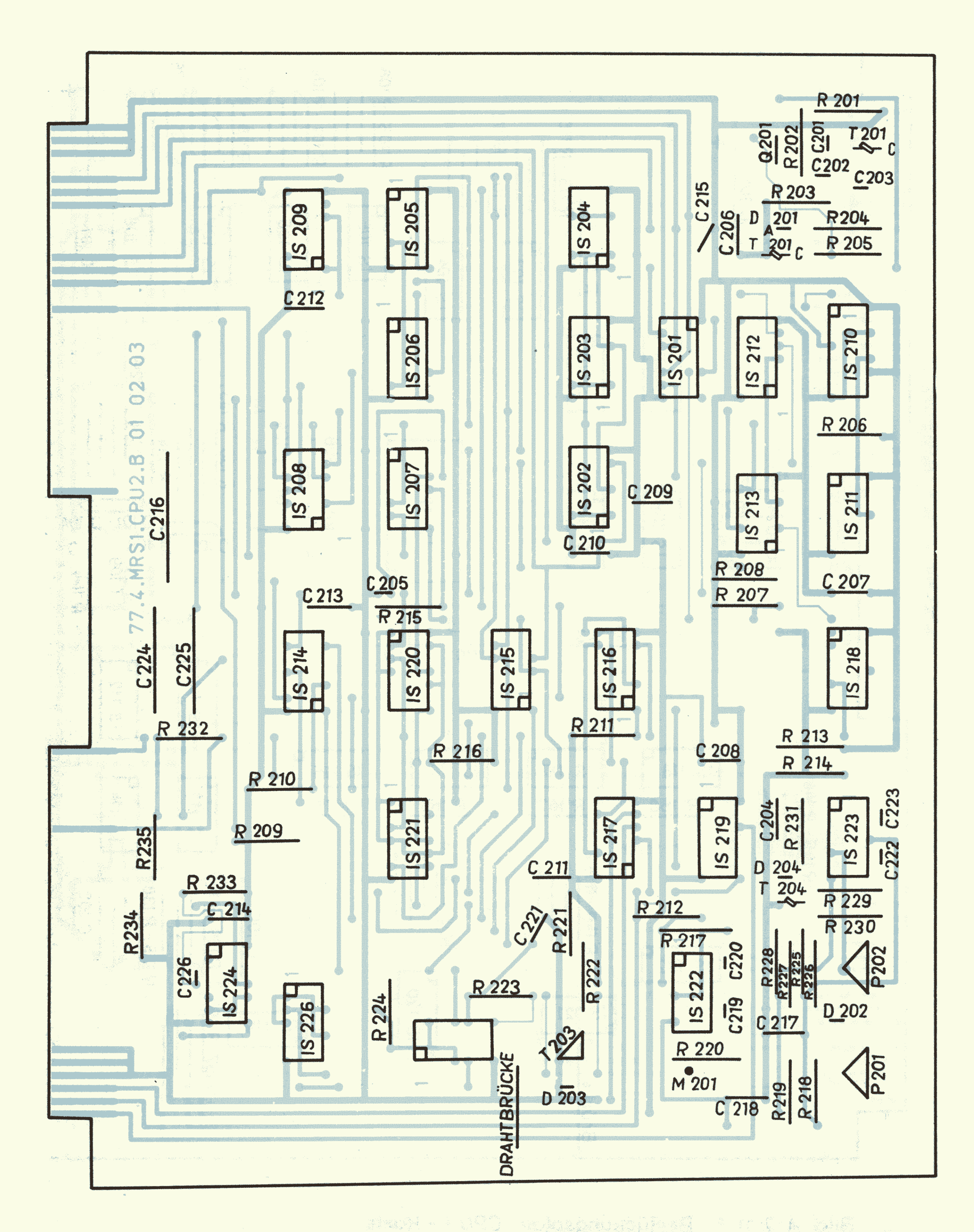


Bild A 2.11. 2. Bestückungsplan CPU 2 - Karte

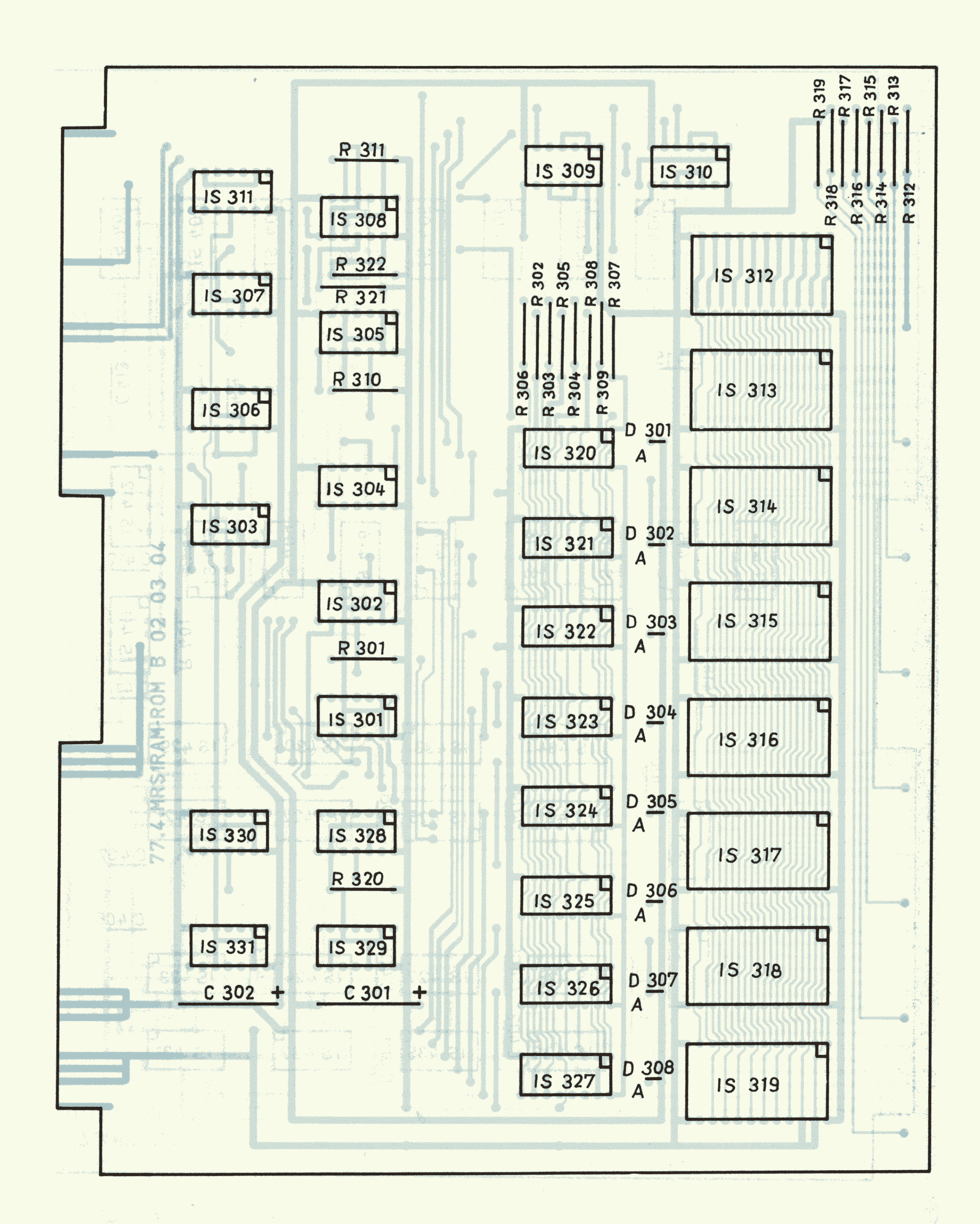


Bild A 2.11.3. Bestückungsplan RAM-ROM Karte

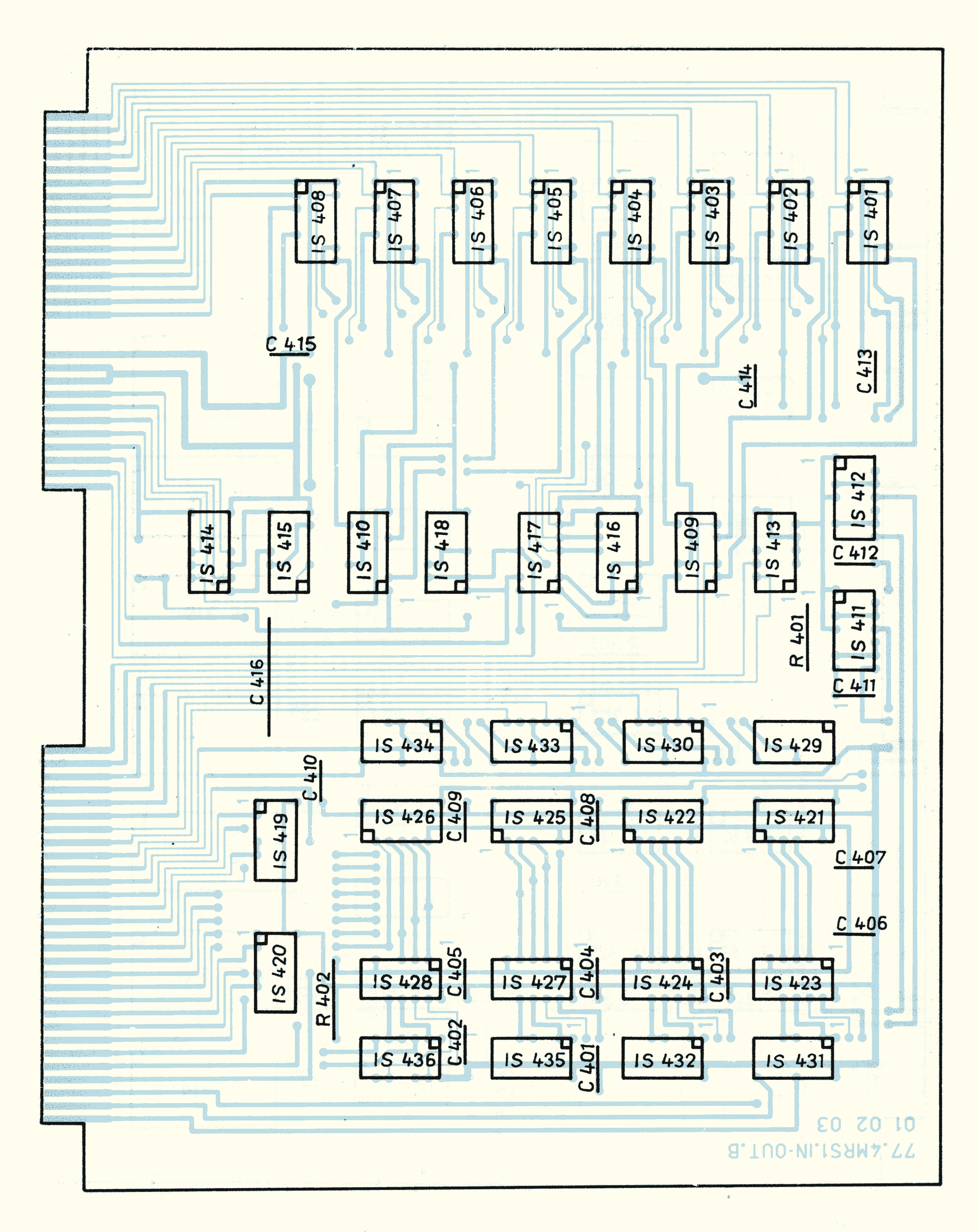


Bild A 2.11.4. Bestückungsplan IN/OUT-Karte

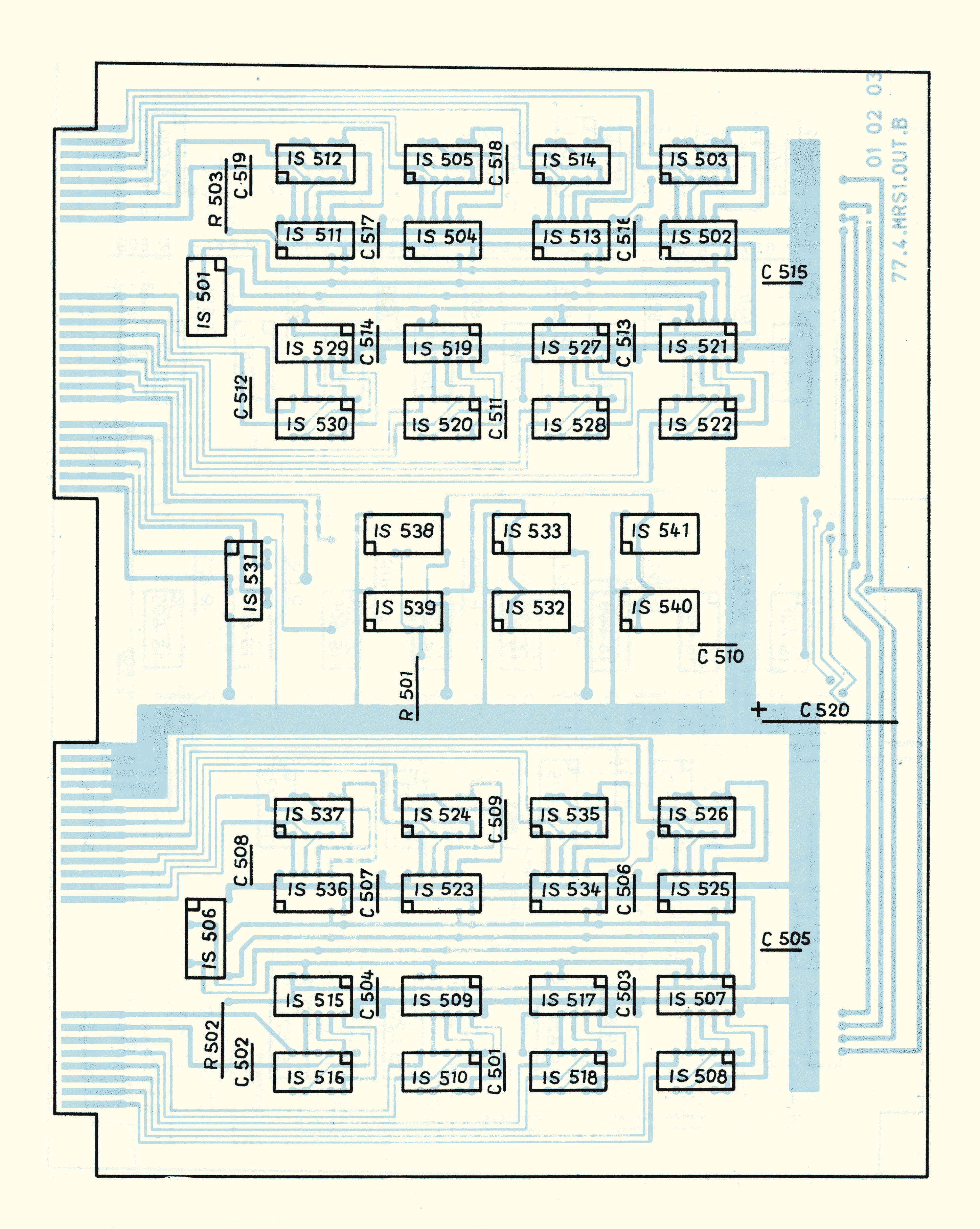


Bild A 2.11.5. Bestückungsplan OUT-Karte

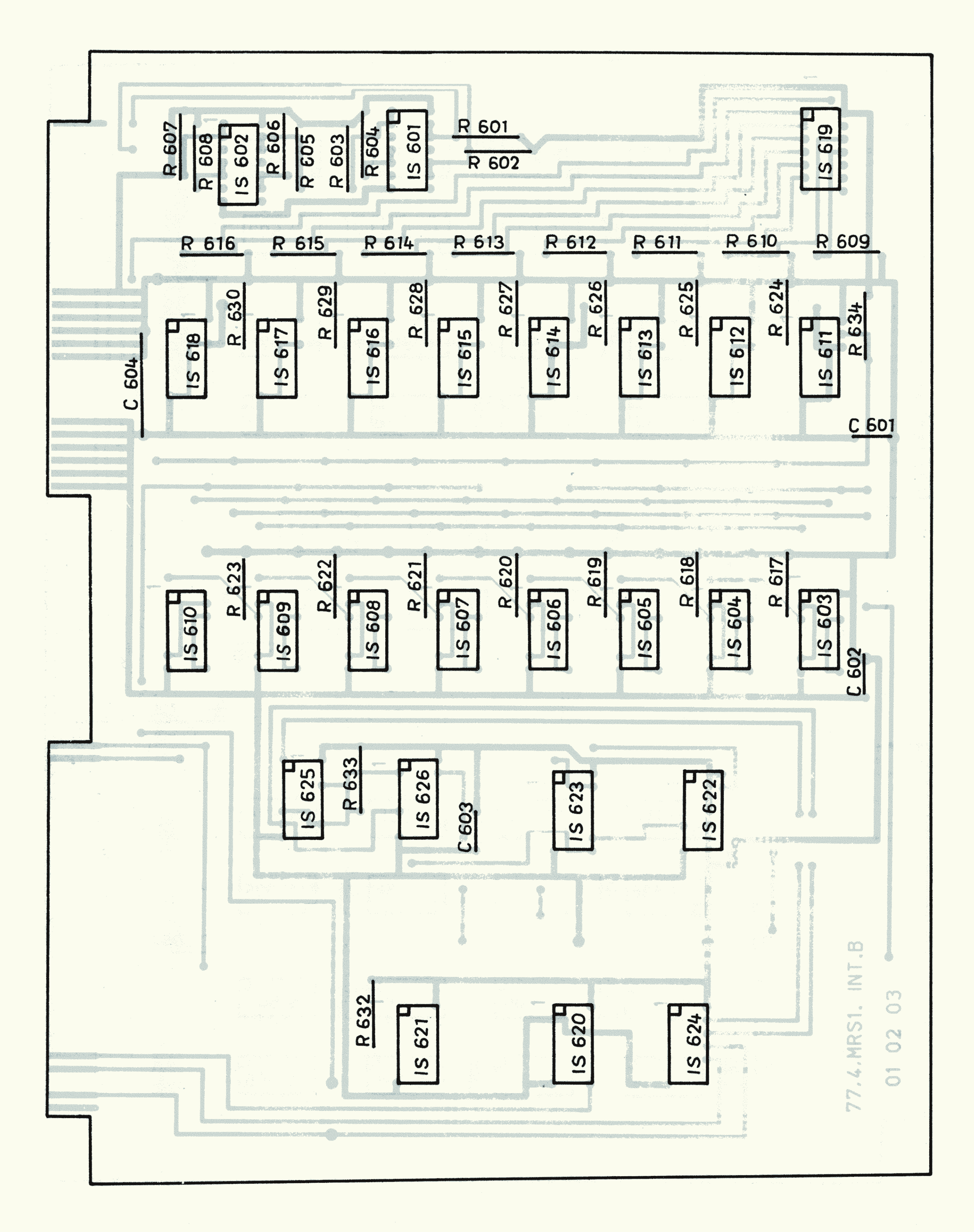


Bild A 2.11.6. Bestückungsplan INT-Karte

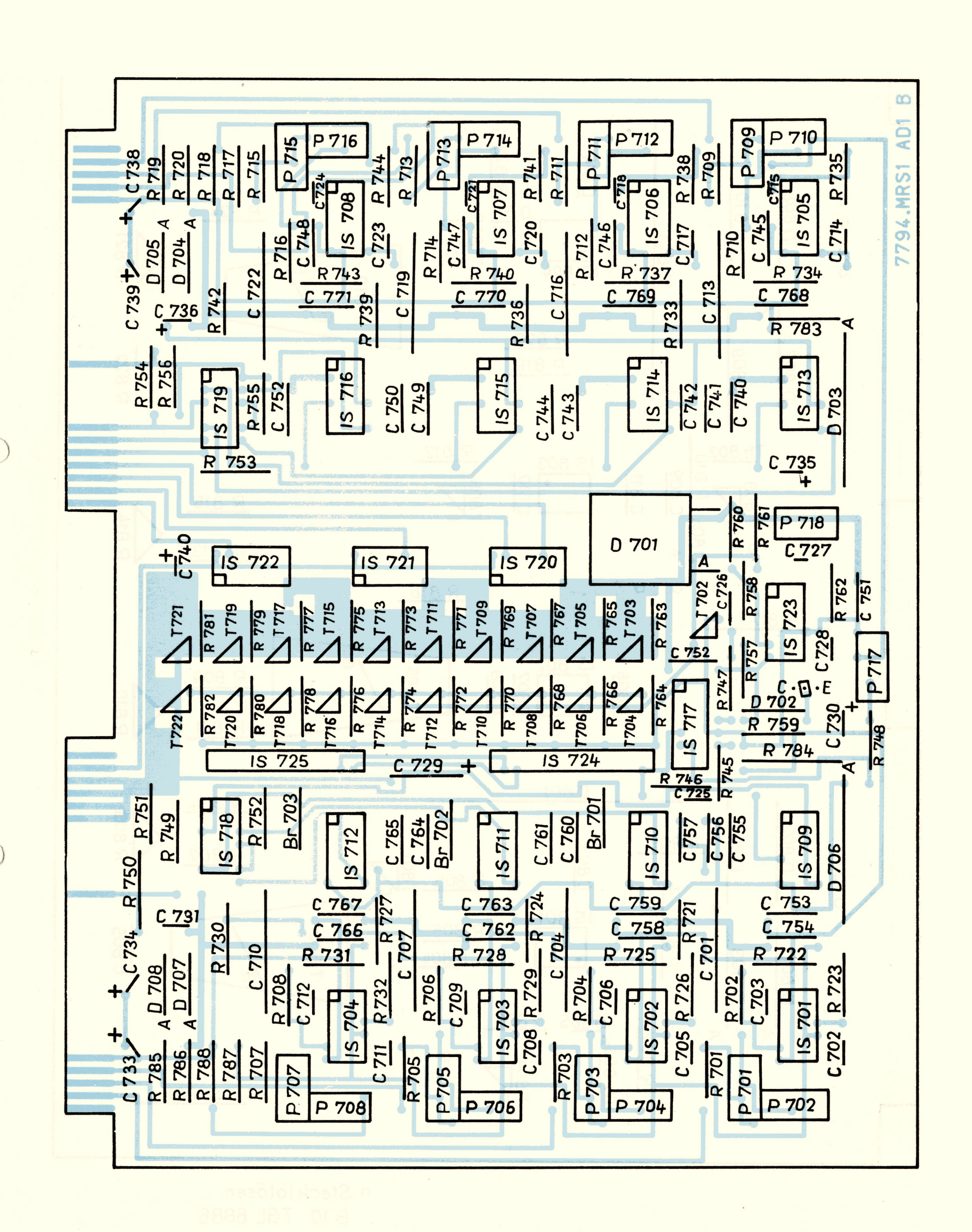


Bild A 2. 11. 7. Bestückungsplan AD - Karte

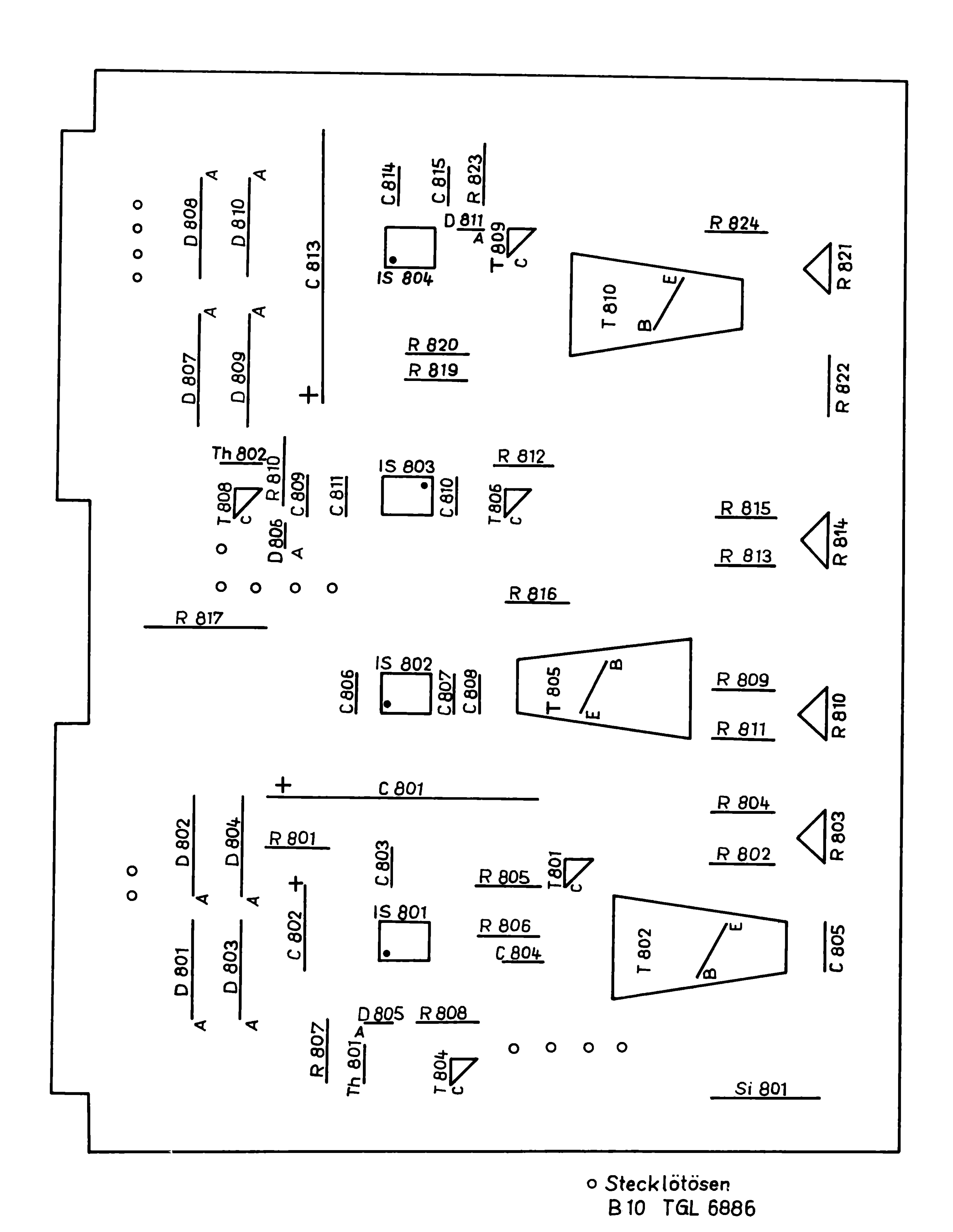


Bild A 2.11.8. Bestückungsplan NT-Karte

Tafel A 2.12. Zuordnung der Speicheradressbereiche

RAM/ ROM-Karte	ROM-Adreßbereich dezimal	ROM-Adreßbereich dual MAD 13 12 11 10 9 8 7 6 5 4 3 2 1 0	ROM-Adreßbereich hexadezimal	Adreßbereich, wo Einsatz von ROM <u>oder</u> RAM wählbar ist: dezimal	dual MAD 13 12 11 10 9 8 7 6 5 4 3 2 1 0	hexadezimal
1	0...1791	0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0	1792...2047	0 0 0 1 1 1 0 0 0 0 0 0 0 0	0 7 0 0
		0 0 0 1 1 0 1 1 1 1 1 1 1 1	0 6 F F		0 0 0 1 1 1 1 1 1 1 1 1 1 1	0 7 F F
2	2048...3839	0 0 1 0 0 0 0 0 0 0 0 0 0 0	0 8 0 0	3840...4095	0 0 1 1 1 1 0 0 0 0 0 0 0 0	0 F 0 0
		0 0 1 1 1 0 1 1 1 1 1 1 1 1	0 E F F		0 0 1 1 1 1 1 1 1 1 1 1 1 1	0 F F F
3	4096...5887	0 1 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 0	5888...6143	0 1 0 1 1 1 0 0 0 0 0 0 0 0	1 7 0 0
		0 1 0 1 1 0 1 1 1 1 1 1 1 1	1 6 F F		0 1 0 1 1 1 1 1 1 1 1 1 1 1	1 7 F F
4	6144...7935	0 1 1 0 0 0 0 0 0 0 0 0 0 0	1 8 0 0	7936...8191	0 1 1 1 1 1 0 0 0 0 0 0 0 0	1 F 0 0
		0 1 1 1 1 0 1 1 1 1 1 1 1 1	1 E F F		0 1 1 1 1 1 1 1 1 1 1 1 1 1	1 F F F
5	8192...9983	1 0 0 0 0 0 0 0 0 0 0 0 0 0	2 0 0 0	9984...10239	1 0 0 1 1 1 0 0 0 0 0 0 0 0	2 7 0 0
		1 0 0 1 1 0 1 1 1 1 1 1 1 1	2 6 F F		1 0 0 1 1 1 1 1 1 1 1 1 1 1	2 7 F F
6	10240...12031	1 0 1 0 0 0 0 0 0 0 0 0 0 0	2 8 0 0	12032...12287	1 0 1 1 1 1 0 0 0 0 0 0 0 0	2 F 0 0
		1 0 1 1 1 0 1 1 1 1 1 1 1 1	2 E F F		1 0 1 1 1 1 1 1 1 1 1 1 1 1	2 F F F
7	12288...14079	1 1 0 0 0 0 0 0 0 0 0 0 0 0	3 0 0 0	14080...14335	1 1 0 1 1 1 0 0 0 0 0 0 0 0	3 7 0 0
		1 1 0 1 1 0 1 1 1 1 1 1 1 1	3 6 F F		1 1 0 1 1 1 1 1 1 1 1 1 1 1	3 7 F F
8	14336...16127	1 1 1 0 0 0 0 0 0 0 0 0 0 0	3 8 0 0	16128...16383	1 1 1 1 1 1 0 0 0 0 0 0 0 0	3 F 0 0
		1 1 1 1 1 0 1 1 1 1 1 1 1 1	3 E F F		1 1 1 1 1 1 1 1 1 1 1 1 1 1	3 F F F

A 2.13. Definition der Steckverbinderlage

Tafel A 2.13.1. Definition der Anschlußbezeichnungen

Anschlußbezeichnung:

LP-Nr./SV-Ebene SV-Seite Pin

Beispiele:

2/2 B 29 ($\overline{\text{WAIT REQ}}$)

32/1 B 26 ($\overline{\text{MAD 11}}$)

Zuordnung der LP-Nr.:

Karte	LP-Nr.
CPU 1	1
CPU 2	2
RAM/ROM (1)	3 bzw. 31
RAM/ROM 2	32
•	•
•	•
•	•
RAM/ROM 8	38
IN/OUT (1)	4 bzw. 41
IN/OUT 2	42
OUT (1)	5 bzw. 51
OUT 2	52
INT	6
AD (1)	7 bzw. 71
AD 2	72
•	
•	
•	
NT	8

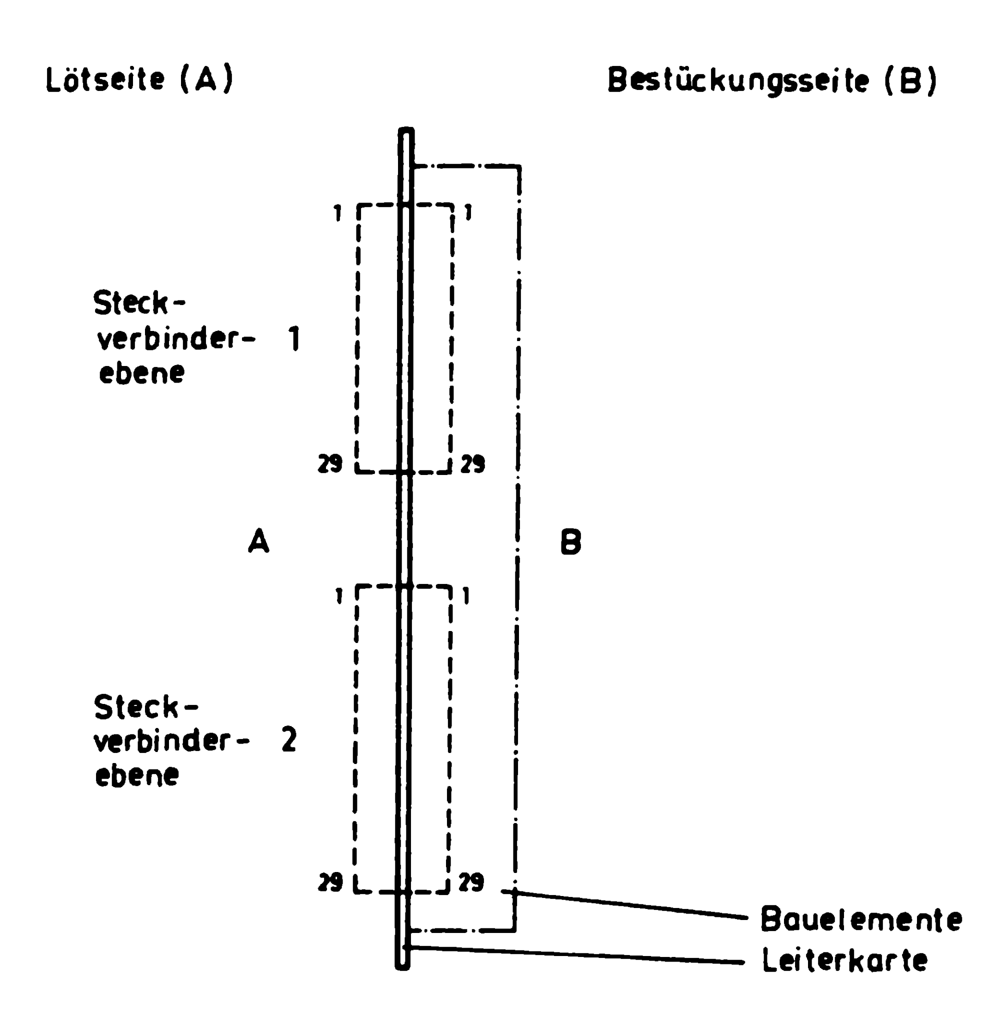


Bild A 2.13.2. Definition der Anschlußbelegung der Leiterkarten

Tafel A 2.14. Wickeldaten des Netztrafos

Netztrafo

Kern	LL 75/25	TGL 0-41 302
Bleche	0,35 E 0,7	TGL 14 342

Wickelkörper A:

W_0 = 517 Wdg.; 0,55 CuL; Lagenisolation nach 4 Lagen

W_1 = 16 Wdg.; 2,0 CuL;

W_2 = 33 Wdg.; 1,1 CuL;

W_3 = 50 Wdg.; 0,55 CuL;

W_4 = 50 Wdg.; 0,55 CuL;

W_5 = 3 Wdg.; 0,55 CuL;

Wickelkörper B wie A.

A 2.15. Anschlußbelegungen der Leiterkarten

Tafel A 2.15.1.: Pinbelegung der CPU 1-Karte

	SV 1		SV 2	
	A	B	A	B
1	MDB 0	U_{CC}	DB 0	
2	MDB 1	U_{CC}	DB 1	
3	MDB 2	U_{CC}	DB 2	
4	MDB 3	U_{CC}	DB 3	
5	MDB 4		DB 4	
6	MDB 5		DB 5	
7	MDB 6	U_{EE}	DB 6	
8	MDB 7	U_{EE}	DB 7	
9	INB 0	U_{EE}		
10	INB 1	U_{EE}		
11	INB 2			
12	INB 3	U_{DD}		
13	INB 4			
14	INB 5	INT	MAD' 0	$\overline{\text{MAD 0}}$
15	INB 6	$\overline{\text{WAIT REQ}}$	MAD' 1	$\overline{\text{MAD 1}}$
16	INB 7	cp 1	MAD' 2	$\overline{\text{MAD 2}}$
17		cp 2	MAD' 3	$\overline{\text{MAD 3}}$
18		sync'	MAD' 4	$\overline{\text{MAD 4}}$
19		s 0	MAD' 5	$\overline{\text{MAD 5}}$
20	IIB 3	s 1	MAD' 6	$\overline{\text{MAD 6}}$
21	IIB 4	s 2	MAD' 7	$\overline{\text{MAD 7}}$
22	IIB 5		MAD' 8	$\overline{\text{MAD 8}}$
23			MAD' 9	$\overline{\text{MAD 9}}$
24			MAD' 10	$\overline{\text{MAD 10}}$
25			MAD' 11	$\overline{\text{MAD 11}}$
26			MAD' 12	$\overline{\text{MAD 12}}$
27	$\overline{\text{DB IN}}$	$\overline{\text{T1}}$	MAD' 13	$\overline{\text{MAD 13}}$
28	$\overline{\text{INT SEL}}$	$\overline{\text{T2}}$	MAD' 14	
29	$\overline{\text{I/O IN SEL}}$	$\overline{\text{DB OUT}}$	MAD' 15	

Tafel A 2.15.2.: Pinbelegung der CPU 2-Karte

	SV 1		SV 2	
	A	B	A	B
1		U_{CC}	sync'	$\overline{\text{DB IN}}$
2	INT	U_{CC}	$\overline{\text{INT SEL}}$	
3		U_{CC}	sync A	
4				
5	$\overline{\text{T2}}$	s 0	$\overline{\text{HLT INT REQ}}$	
6	$\overline{\text{T3 W}}$	s 1		
7	$\overline{\text{T3 S}}$	s 2		U_5
8				
9	$\overline{\text{T3}}$		$\overline{\text{INT REQ}}$	
10			$\overline{\text{INT REQ LTH}}$	
11	$\overline{\text{I/O OUT}}$	$\overline{\text{T1}}$		
12		$\overline{\text{T1 I}}$		
13		$\overline{\text{I/O IN SEL}}$	$\overline{\text{CI}}$	
14			$\overline{\text{CLE}}$	
15		$\text{R/}\overline{\text{W}}$	CLA	
16	$\overline{\text{T3 A}}$			
17				
18	MAD' 12			
19	MAD' 13			
20	$\overline{\text{PCI}}$			
21	$\overline{\text{PCR}}$			
22	$\overline{\text{PCW}}$			
23	$\overline{\text{PCC}}$			
24	MAD' 14			U_{EE}
25	MAD' 15			U_{EE}
26				U_{EE}
27	cp 2		$\overline{\text{DB OUT}}$	$\overline{\text{HOLD ACK}}$
28	cp 1		U_4	$\overline{\text{HOLD REQ}}$
29			U_3	$\overline{\text{WAIT REQ}}$

Tafel A 2.15.3. Pinbelegung der RAM/ROM-Karte

	SV 1		SV 2	
	A	B	A	B
1	COL SEL 0		MAD 2	U_{DD}
2	COL SEL 1		MAD 1	U_{DD}
3	COL SEL 3		MAD 3	U_{DD}
4	COL SEL 2		MAD 0	
5			MAD 4	
6			$\overline{\text{MAD 7}}$	
7			$\overline{\text{MAD 5}}$	
8	$\overline{\text{WS}}$		$\overline{\text{MAD 2}}$	
9	R/$\overline{\text{W}}$		$\overline{\text{MAD 6}}$	
10	$\overline{\text{MAD 12}}$		$\overline{\text{MAD 1}}$	
11		MCS	$\overline{\text{MAD 0}}$	
12	CARD SEL 2		$\overline{\text{MAD 4}}$	
13	CARD SEL 3		$\overline{\text{MAD 3}}$	
14	CARD SEL 1		MDB 0	
15	CARD SEL 0		MDB 1	
16		LINE SEL' B	MDB 2	
17		LINE SEL' A	MDB 3	
18	$\overline{\text{MAD 13}}$		MDB 4	
19	$\overline{\text{MAD 10}}$		MDB 5	
20			MDB 6	U_{EE}
21	LINE SEL		MDB 7	U_{EE}
22	LINE SEL		DB 0	U_{EE}
23	LINE SEL		DB 1	
24	LINE SEL		DB 2	
25	$\overline{\text{MAD 8}}$		DB 3	U_{CC}
26	$\overline{\text{MAD 9}}$	$\overline{\text{MAD 11}}$	DB 4	U_{CC}
27	MAD 5		DB 5	U_{CC}
28	MAD 7		DB 6	
29	MAD 6		DB 7	

Tafel A 2.15.4.: Pinbelegung der IN/OUT-Karten

	SV 1		SV 2	
	A	B	A	B
1	IN 2/6-0	IN 0/4-0	U_{CC}	INB
2	IN 1/5-0	IN 3/1-0	U_{CC}	INB
3	IN 2/6-1	IN 0/4-1	U_{CC}	I/O OUT
4	IN 1/5-1	IN 3/1-1	OUT 8/12-0	$\overline{\text{I/O OUT}}$
5	IN 2/6-2	IN 0/4-2	OUT 8/12-1	OUT 8/12-4
6	IN 1/5-2	IN 3/7-2	OUT 8/12-2	OUT 8/12-5
7	IN 2/6-3	IN 0/4-3	OUT 8/12-3	OUT 10/14-0
8	IN 1/5-3	IN 3/7-3	OUT 8/12-6	OUT 10/14-1
9	IN 2/6-4	IN 0/4-4	OUT 8/12-7	OUT 10/14-6
10	IN 1/5-4	IN 3/7-4	OUT 10/14-2	OUT 10/14-7
11	IN 2/6-5	IN 0/4-5	OUT 10/14-3	$\overline{\text{MAD' 2}}$
12	IN 1/5-5	IN 3/7-5	OUT 10/14-4	MAD' 0
13	IN 2/6-6	IN 0/4-6	OUT 10/14-5	$\overline{\text{MAD' 0}}$
14	IN 1/5-6	IN 3/7-6	U_{EE}	MAD' 1
15	IN 2/6-7	IN 0/4-7	U_{EE}	$\overline{\text{MAD' 1}}$
16	IN 1/5-7	IN 3/7-7	U_{EE}	MAD' 2
17			OUT 11/15-7	MAD' 3
18			OUT 11/15-6	$\overline{\text{MAD' 3}}$
19	INB 3	U_{CC}	OUT 11/15-5	MAD' 4
20	INB 2	U_{EE}	OUT 11/15-4	MAD' 5
21	$\overline{\text{PORT SEL}}$ 2/6	U_{EE}	OUT 11/15-3	MAD' 6
22	$\overline{\text{PORT SEL}}$ 0/4	OCS 1	OUT 11/15-2	$\overline{\text{MAD' 6}}$
23	$\overline{\text{PORT SEL}}$ 1/5	OCS 0	OUT 11/15-1	MAD' 7
24	$\overline{\text{PORT SEL}}$ 3/7	OCS 2	OUT 11/15-0	$\overline{\text{MAD' 7}}$
25	INB	MAD' 11	OUT 9/13-7	$\overline{\text{MAD' 5}}$
26	INB	MAD'' 11	OUT 9/13-6	$\overline{\text{MAD' 4}}$
27	INB	MAD' 12	OUT 9/13-5	OUT 9/13-2
28	INB	MAD' 9	OUT 9/13-4	OUT 9/13-1
29	MAD' 13	MAD' 10	OUT 9/13-3	OUT 9/13-0

(Definition der Lage der Ports: IN x/y-z (bzw. OUT x/y-z);
IN x-z (bzw. OUT x-z) auf Karte IN/OUT 1 [41] ,
IN y-z (bzw. OUT y-z) auf Karte IN/OUT 2 [42] .)

Tafel A 2.15.5.: Pinbelegung der OUT-Karten

	SV 1		SV 2	
	A	B	A	B
1	OUT 20/28-0	OUT 20/28-2	U_{EE}	U_{CC}
2	OUT 20/28-1	OUT 20/28-3	U_{EE}	U_{CC}
3	OUT 17/25-0	OUT 17/25-2	U_{EE}	U_{CC}
4	OUT 17/25-1	OUT 17/25-3	U_{EE}	U_{CC}
5	OUT 16/24-0	OUT 16/24-2	U_{EE}	U_{CC}
6	OUT 16/24-1	OUT 16/24-3	OUT 18/26-4	OUT 18/26-6
7	OUT 21/29-0	OUT 21/29-2	OUT 18/26-5	OUT 18/26-7
8	OUT 21/29-1	OUT 21/29-3	OUT 23/31-4	OUT 23/31-6
9			OUT 23/31-5	OUT 23/31-7
10			OUT 22/30-4	OUT 22/30-6
11			OUT 22/30-5	OUT 22/30-7
12			OUT 19/27-4	OUT 19/27-6
13			OUT 19/27-5	OUT 19/27-7
14	MAD' 0	OUT 19/27-3		
15	MAD' 1	OUT 19/27-2		
16	MAD' 2	OUT 22/30-3		
17	MAD' 3	OUT 22/30-2		
18	OUT 19/27-1	OUT 23/31-3	MAD' 4	
19	OUT 19/27-0	OUT 23/31-2	MAD' 5	
20	OUT 22/30-1	OUT 18/26-3	MAD' 6	
21	OUT 22/30-0	OUT 18/26-2	MAD' 7	
22	OUT 23/31-1	MAD' 11	OUT 21/29-5	OUT 21/29-7
23	OUT 23/31-0		OUT 21/29-4	OUT 21/29-6
24	OUT 18/26-1	CARD SEL 1/2	OUT 16/24-5	OUT 16/24-7
25	OUT 18/26-0	$\overline{\text{I/O OUT}}$	OUT 16/24-4	OUT 16/24-6
26		$\overline{\text{PORT SEL 3/7}}$	OUT 17/25-5	OUT 17/25-7
27		$\overline{\text{PORT SEL 2/6}}$	OUT 17/25-4	OUT 17/25-6
28		$\overline{\text{PORT SEL 1/5}}$	OUT 20/28-5	OUT 20/28-7
29		$\overline{\text{PORT SEL 0/4}}$	OUT 20/28-4	OUT 20/28-6

(Definition der Lage der OUT-Ports: OUT x/y-z;
OUT x-z auf Karte OUT 1 [51] ,
OUT y-z auf Karte OUT 2 [52] .)

Tafel A 2.15.6. Pinbelegung der INT-Karte

	SV 1		SV 2	
	A	B	A	B
1	IA 7			$\overline{\text{INT SEL}}$
2				$\overline{\text{T3A}}$
3	IA 6			
4	IA 5			
5			$\overline{\text{T2}}$	
6	IA 4			
7	IA 3			
8	IA 2			
9	IA 1		$\overline{\text{T3}}$	
10	IA 0		IIB 3	
11			U_{EE}	
12			U_{EE}	
13			U_{EE}	
14		U_{CC}	IIB 4	
15		U_{CC}	IIB 5	
16		U_{CC}	T3A	
17		U_{CC}	$\overline{\text{IEND}}$	
18		U_{CC}		
19		U_{CC}		
20				
21				
22				
23			CLK	
24		U_{EE}	HIGH	
25		U_{EE}		INT DEL
26		U_{EE}		$\overline{\text{CLA}}$
27		U_{EE}		
28	INT INH	U_{EE}		$\overline{\text{INT T2}}$
29	$\overline{\text{INT REQ}}$	U_{EE}		

Tafel A 2.15.7.: Pinbelegung der AD-Karte

	SV 1		SV 2	
	A	B	A	B
1				
2	AE 8	AE 5		U_{CC}
3		AE 6	DA 4	U_{EE}
4		AE 7	U_{CC}	U_{EE}
5		U_4	DA 3	U_{EE}
6	U_{EE}		SEL 2	U_{EE}
7	U_3			
8			DA 1	
9			DA 2	
10				
11				
12				
13				
14			U_{EE}	
15				
16	DA 7			
17				
18	DA 8	U_{CC}		
19	SEL 1			
20				
21				
22				
23	DE 7	DA 6	U_3	
24	DE 6	DA 5		
25	DE 4	U_{EE}		U_4
26	DE 3	DE 9		AE 3
27	DE 1	DE 8		AE 2
28	DE 0	DE 5	AE 4	AE 1
29		DE 2	AA	

Tafel A 2.15.8. Pinbelegung der NT-Karte

	SV 1		SV 2	
	A		A	
1	U_4			
2	U_4			
3				
4				
5	U_{5a}			
6	U_{5a}			
7	U_{5b}			
8	U_{5b}			
9				
10			U_3	
11			U_3	
12				
13	U_{DD}			
14	U_{DD}			
15	$U_{EEfühl}$		$U_{CCfühl}$	
16	$U_{EEfühl}$		$U_{CCfühl}$	
17	U_{EE}		U_{CC}	
18	U_{EE}		U_{CC}	
19	U_{EE}		U_{CC}	
20	U_{EE}		U_{CC}	
21	U_{EE}		U_{CC}	
22	U_{EE}		U_{CC}	
23	U_{EE}		U_{CC}	
24	U_{EE}		U_{CC}	
25	U_{EE}		U_{CC}	
26	U_{EE}		U_{CC}	
27	U_{EE}		U_{CC}	
28	U_{EE}		U_{CC}	
29	U_{EE}		U_{CC}	

A 2.16. Verdrahtungslisten

Tafel A 2.16.1. Verbindungen zur Stromversorgung

Verbindungen			Bemerkungen
1/1B1...4	-	8/2A15...17	U_{CC}
1/1B7...10	-	8/1A15...17	U_{EE}
1/1B12	-	8/1A13	U_{DD}
2/1B1...3	-	8/2A18...19	U_{CC}
2/2B24...26	-	8/1A18...19	U_{EE}
2/2A29	-	8/2A10	U_3
2/2A28	-	8/1A1	U_4
2/2B7	-	8/1A5	U_{5a}
2/2B24	-	8/1A7	U_{5b}
3/2B25...27	-	8/2A19...22	U_{CC}
3/2B20...22	-	8/1A19...22	U_{EE}
3/2B1...3	-	8/1A13...14	U_{DD}
4/2A1...3	-	8/2A23...25	U_{CC}
4/2A14...16	-	8/1A23...25	U_{EE}
5/2B1...5	-	8/2A26...29	U_{CC}
5/2A1...5	-	8/1A26...29	U_{EE}
6/1B14...19	-	8/2A23	U_{CC}
6/1B24...29	-	8/1A23	U_{EE}

Tafel A 2.16.2. Verdrahtung der CPU-Baugruppe

Verbindungen			Bemerkungen
2/1A2	-	1/1B14	INT
2/1A28	-	1/1B16	cp 1
2/1A27	-	1/1B17	cp 2
1/1B18	-	2/2A1	sync'
1/1B19	-	2/1B5	s0
1/1B20	-	2/1B6	s1
1/1B21	-	2/1B7	s2
2/1B11	-	1/1B27	$\overline{\text{T1}}$
2/1A5	-	1/1B28	$\overline{\text{T2}}$
1/2A26	-	2/1A18	MAD' 12
1/2A27	-	2/1A19	MAD' 13
1/2A28	-	2/1A24	MAD' 14
1/2A29	-	2/1A25	MAD' 15
2/2B1	-	1/1A27	$\overline{\text{DB IN}}$
2/2A27	-	1/1B29	$\overline{\text{DB OUT}}$
2/2A2	-	1/1A28	$\overline{\text{INT SEL}}$
2/1B13	-	1/1A29	$\overline{\text{I/O IN SEL}}$
2/2B29	-	1/1B15	$\overline{\text{WAIT REQ}}$

A 2.16.3. Verdrahtung der RAM/ROM-Baugruppe

Tafel A 2.16.3.1. Spezifikation der RAM/ROM-Karten

Verbindungen			Bemerkungen
31/1B11	-	31/1A15	CARD SEL 0
31/1B17	-	31/1A24	LINE SEL 0
31/1B16	-	31/1A23	LINE SEL 1
32/1B11	-	31/1A15	CARD SEL 0
32/1B17	-	31/1A21	LINE SEL 2
32/1B16	-	31/1A22	LINE SEL 3
33/1B11	-	31/1A14	CARD SEL 1
33/1B17	-	31/1A24	LINE SEL 0
33/1B16	-	31/1A23	LINE SEL 1
34/1B11	-	31/1A14	CARD SEL 1
34/1B17	-	31/1A21	LINE SEL 2
34/1B16	-	31/1A22	LINE SEL 3
35/1B11	-	31/1A12	CARD SEL 2
35/1B17	-	31/1A24	LINE SEL 0
35/1B16	-	31/1A23	LINE SEL 1
36/1B11	-	31/1A12	CARD SEL 2
36/1B17	-	31/1A21	LINE SEL 2
36/1B16	-	31/1A22	LINE SEL 3
37/1B11	-	31/1A13	CARD SEL 3
37/1B17	-	31/1A24	LINE SEL 0
37/1B16	-	31/1A23	LINE SEL 1
38/1B11	-	31/1A13	CARD SEL 3
38/1B17	-	31/1A21	LINE SEL 2
38/1B16	-	31/1A22	LINE SEL 3

Tafel A 2.16.3.2. Bus- und Signalverbindungen

Verbindungen	Bemerkungen
31/2A11 - 35/2A11	$\overline{\text{MAD 0}}$
31/2A10 - 35/2A10	$\overline{\text{MAD 1}}$
⋮	⋮
31/2A6 - 35/2A6	$\overline{\text{MAD 7}}$
31/2A4 - 32/2A4 - ... - 34/2A4	MAD 0
31/2A2 - 32/2A2 - ... - 34/2A2	MAD 1
⋮	⋮
31/1A28 - 32/1A28 - ... - 34/1A28	MAD 7
35/2A4 - 36/2A4 - ... - 38/2A4	MAD 0
35/2A2 - 36/2A2 - ... - 38/2A2	MAD 1
⋮	⋮
35/1A28 - 36/1A28 - ... - 38/1A28	MAD 7
31/1A1 - 32/1A1 - ... - 34/1A1	COL SEL 0
31/1A2 - 32/1A2 - ... - 34/1A2	COL SEL 1
31/1A4 - 32/1A4 - ... - 34/1A4	COL SEL 2
31/1A3 - 32/1A3 - ... - 34/1A3	COL SEL 3
35/1A1 - 36/1A1 - ... - 38/1A1	COL SEL 0
35/1A2 - 36/1A2 - ... - 38/1A2	COL SEL 1
35/1A4 - 36/1A4 - ... - 38/1A4	COL SEL 2
35/1A3 - 36/1A3 - ... - 38/1A3	COL SEL 3
31/1A25 - 35/1A25	$\overline{\text{MAD 8}}$
31/1A26 - 35/1A36	$\overline{\text{MAD 9}}$
31/1A8 - 32/1A8 - ... - 38/1A8	$\overline{\text{WS}}$
31/1A9 - 32/1A9 - ... - 38/1A9	R/$\overline{\text{W}}$
31/2A14 - 32/2A14 - ... - 38/2A14	MDB 0
31/2A15 - 32/2A15 - ... - 38/2A15	MDB 1
⋮	⋮
31/2A21 - 32/2A21 - ... - 38/2A21	MDB 7
31/2A22 - 32/2A22 - ... - 38/2A22	DB 0
31/2A23 - 32/2A23 - ... - 38/2A23	DB 1
⋮	⋮
31/2A29 - 32/2A29 - ... - 38/2A29	DB 7

Tafel A 2.16.4. Verdrahtung der Ein-/Ausgabebaugruppe

Verbindungen	Bemerkungen
41/1A22 - 42/1A22 - 51/1B29 - 52/1B29	$\overline{\text{PORT SEL 0/4}}$
41/1A23 - 42/1A23 - 51/1B28 - 52/1B28	$\overline{\text{PORT SEL 1/5}}$
41/1A21 - 42/1A21 - 51/1B27 - 52/1B27	$\overline{\text{PORT SEL 2/6}}$
41/1A24 - 42/1A24 - 51/1B26 - 52/1B26	$\overline{\text{PORT SEL 3/7}}$
41/2B2 - 42/2B2	INB 0
41/2B1 - 42/2B1	INB 1
⋮	⋮
41/1A27 - 42/1A27	INB 7
41/2B12 - 51/1A14 - 52/1A14	MAD' 0
41/2B14 - 51/1A15 - 52/1A15	MAD' 1
⋮	⋮
41/2B23 - 51/2A21 - 52/2A21	MAD' 7
41/2B13 - 42/2B13	$\overline{\text{MAD' 0}}$
41/2B15 - 42/2B15	$\overline{\text{MAD' 1}}$
⋮	⋮
41/2B24 - 42/2B24	$\overline{\text{MAD' 7}}$
41/1B25 - 51/1B22 - 52/1B22	MAD' 11
41/1B26 - 42/1B25	MAD'' 11
41/2B4 - 42/2B4 - 51/1B25 - 52/1B25	$\overline{\text{I/O OUT}}$
41/1B23 - 42/1B23	OCS 0
41/1B22 - 51/1B24	OCS 1
41/1B24 - 52/1B24	OCS 2

Tafel A 2.16.5. Verbindungen zwischen CPU- und RAM/ROM-Baugruppe

Verbindungen	Bemerkungen
1/2B14 - 3/2A11	$\overline{\text{MAD 0}}$
1/2B15 - 3/2A10	$\overline{\text{MAD 1}}$
1/2B16 - 3/2A8	$\overline{\text{MAD 2}}$
1/2B17 - 3/2A13	$\overline{\text{MAD 3}}$
1/2B18 - 3/2A12	$\overline{\text{MAD 4}}$
1/2B19 - 3/2A7	$\overline{\text{MAD 5}}$
1/2B20 - 3/2A9	$\overline{\text{MAD 6}}$
1/2B21 - 3/2A6	$\overline{\text{MAD 7}}$
1/2B22 - 3/1A25	$\overline{\text{MAD 8}}$
1/2B23 - 3/1A26	$\overline{\text{MAD 9}}$
1/2B24 - 3/1A19	$\overline{\text{MAD 10}}$
1/2B25 - 3/1B26	$\overline{\text{MAD 11}}$
1/2B26 - 3/1A10	$\overline{\text{MAD 12}}$
1/2B27 - 3/1A18	$\overline{\text{MAD 13}}$
1/2A1 - 3/2A22	DB 0
1/2A2 - 3/2A23	DB 1
1/2A3 - 3/2A24	DB 2
1/2A4 - 3/2A25	DB 3
1/2A5 - 3/2A26	DB 4
1/2A6 - 3/2A27	DB 5
1/2A7 - 3/2A28	DB 6
1/2A8 - 3/2A29	DB 7
3/2A14 - 1/1A1	MDB 0
3/2A15 - 1/1A2	MDB 1
3/2A16 - 1/1A3	MDB 2
3/2A17 - 1/1A4	MDB 3
3/2A18 - 1/1A5	MDB 4
3/2A19 - 1/1A6	MDB 5
3/2A20 - 1/1A7	MDB 6
3/2A21 - 1/1A8	MDB 7
2/1B15 - 3/1A9	R/$\overline{\text{W}}$

Tafel A 2.16.6. Verbindungen zwischen CPU- und Ein-/Ausgabebaugruppe

Verbindungen	Bemerkungen
1/2A14 - 4/2B12	MAD' 0
1/2A15 - 4/2B14	MAD' 1
1/2A16 - 4/2B16	MAD' 2
1/2A17 - 4/2B17	MAD' 3
1/2A18 - 4/2B19	MAD' 4
1/2A19 - 4/2B20	MAD' 5
1/2A20 - 4/2B21	MAD' 6
1/2A21 - 4/2B23	MAD' 7
1/2A23 - 4/1B28	MAD' 9
1/2A24 - 4/1B29	MAD' 10
1/2A25 - 4/1B25	MAD' 11
1/2A26 - 4/1B27	MAD' 12
1/2A27 - 4/1A29	MAD' 13
4/2B2 - 1/1A9	INB 0
4/2B1 - 1/1A10	INB 1
4/1A20 - 1/1A11	INB 2
4/1A19 - 1/1A12	INB 3
4/1A25 - 1/1A13	INB 4
4/1A26 - 1/1A14	INB 5
4/1A28 - 1/1A15	INB 6
4/1A27 - 1/1A16	INB 7
2/1A11 - 4/2B4	$\overline{\text{I/O OUT}}$

A 2.16.7. Verbindungen zwischen CPU- und INT-Baugruppe

Verbindungen	Bemerkungen	
2/1A5 - 6/2A5	$\overline{\text{T 2}}$	
2/1A9 - 6/2A9	$\overline{\text{T 3}}$	
2/1A16 - 6/2B2	$\overline{\text{T 3 A}}$	
6/1A29 - 2/2A9	$\overline{\text{INT REQ}}$	siehe 2.4.2.10.
- 2/2A5	$\overline{\text{HLT INT REQ}}$	siehe 2.4.2.10.
2/2A15 - 6/2B26	$\overline{\text{CLA}}$	
2/2A2 - 6/2B1	$\overline{\text{INT SEL}}$	
6/2A10 - 1/1A20	IIB 3	
6/2A14 - 1/1A21	IIB 4	
6/2A15 - 1/1A22	IIB 5	
6/2A23 - 6/2A16	CLK, T3A	siehe 2.7.3.
6/1A28 - 6/2B25	INT INH	siehe 2.7.3.
6/2A17 - [OUT x-y]	$\overline{\text{IEND}}$	siehe 2.7.3.

Tafel A 2.16.8. Anschluß der AD-Baugruppe an den µR

Verbindungen		Bemerkungen
7/1B18	- 8/2A20	U_{CC}
7/2A4	- 8/2A21	U_{CC}
7/2B2	- 8/2A22	U_{CC}
7/1A6	- 8/1A20	U_{EE}
7/1B25	- 8/1A21	U_{EE}
7/2A14	- 8/1A22	U_{EE}
7/2B3...6	- 8/1A23	U_{EE}
7/1A7	- 8/2A10	U_3
7/2A23	- 8/2A11	U_3
7/1B5	- 8/1A1	U_4
7/2B25	- 8/1A2	U_4
7/2A8	- (IN x-a)	DA 1
7/2A9	- (IN x-b)	DA 2
7/2A5	- (IN x-c)	DA 3
7/2A3	- (IN x-d)	DA 4
7/1B24	- (IN x-e)	DA 5
7/1B23	- (IN x-f)	DA 6
7/1A16	- (IN x-g)	DA 7
7/1A18	- (IN x-h)	DA 8
7/1A28	- (OUT y-a)	DE 0
7/1A27	- (OUT y-b)	DE 1
7/1B29	- (OUT y-c)	DE 2
7/1A26	- (OUT y-d)	DE 3
7/1A25	- (OUT y-e)	DE 4
7/1B28	- (OUT y-f)	DE 5
7/1A24	- (OUT y-g)	DE 6
7/1A23	- (OUT y-h)	DE 7
7/1B27	- (OUT z-a)	DE 8
7/1B26	- (OUT z-b)	DE 9
7/1A19	- (OUT z-c)	SEL 1
7/2A6	- (OUT z-d)	SEL 2

A 2.17. Stücklisten zu den Leiterkarten

Tafel A 2.17.1.: Stückliste CPU 1-Karte

Schaltkreise

IS 101...IS 108, IS 113, IS 114, IS 119...IS 122	D 103	TGL 27148
IS 109	D 100	TGL 26152
IS 110...IS 112	D 204	TGL 29263
IS 115...IS 118	D 195	TGL 28467
IS 123	U 808 D	TGL 32058

Transistoren

T 101...T 109	KT 326 B

Dioden

D 101-D 104, D 105-D 108	Mehrfachdiode	SAM 64	TGL 25546

Kondensatoren

C 101, C 102	47 nF	Kondensator SDVU 47 nF/50/63 7619.84
C 103...C 105	100/6	Tantal-Kondensatoren 100/6 TGL 200-8519

Widerstände

R 101...R 125, R 143, R 163...R 168	1 kOhm	5 %	250.207 TK	TGL 8728
R 126...R 134	3 kOhm	5 %	250.207 TK	TGL 8728
R 135...R 142	820 Ohm	5 %	250.207 TK	TGL 8728
R 144	240 Ohm	5 %	250.207 TK	TGL 8728
R 145...R 160	2 kOhm	5 %	250.207 TK	TGL 8728
R 161, R 162	20 kOhm	5 %	250.207 TK	TGL 8728

Tafel A 2.17.2.: Stückliste CPU 2-Karte

Schaltkreise

IS 201, IS 207	D 204	TGL 29263
IS 202...IS 204	D 120	TGL 26152
IS 205, IS 206	D 150	TGL 26153
IS 208, IS 209, IS 215, IS 218, IS 219, IS 224	D 100	TGL 26152
IS 210	D 192	TGL 29267
IS 211, IS 216, IS 217, IS 220, IS 221, IS 225, IS 226	D 174	TGL 29266
IS 212, IS 213	D 172	TGL 27977
IS 214	D 110	TGL 26152
IS 222, IS 223	A 109	TGL 28873

Transistoren

T 201, T 202, T 204	SS 216 D	TGL 26818
T 203	SF 126 E	TGL 11811

Dioden

D 201, D 203, D 204	SAY 32	TGL 200-8466
D 202	SZX 21/5,6	TGL 27338

Kondensatoren

C 201	47	EDVU-Kondensator N 470	TGL 24100
C 202	82	EDVU-Kondensator N 750	TGL 24100
C 203	1n	EDVU-Kondensator	
C 204	100n	SDVU 100 nF/50/63	78 19.84
C 205	330	EDVU-Kondensator	
C 206...C 214, C 217, C 218	47n	SDVU 47 nF/50/63	76 19.84
C 215, C 221	22/40	Elektrolytkondensator	TGL 26629
C 216	100/6	Elektrolytkondensator	
C 219, C 222	3,3	EDVU-Kondensator P 100	TGL 24100
C 220, C 223	10	EDVU-Kondensator N 150	TGL 24100
C 224, C 225	1 µ 5	MKL 3-Kondensator 1,5 µ/63	TGL 10793
C 226	2,2 n	EDVU-Kondensator P 100	TGL 24100

Einstellregler

P 201, P 202	1 kOhm	0,1 W		TGL 11886

Widerstände

R 201, R 202, R 232	10 kOhm	5 %	250.207 TK	TGL 8728
R 203, R 204	820	5 %	250.207 TK	TGL 8728
R 205, R 220, R 230	2,2 kOhm	5 %	250.207 TK	TGL 8728
R 206...R 212, R 216, R 219, R 224, R 231	1 kOhm	5 %	250.207 TK	TGL 8728
R 213, R 214, R 233, R 234	6,2 kOhm	5 %	250.207 TK	TGL 8728
R 215	220	5 %	250.207 TK	TGL 8728
R 217	330	5 %	250.207 TK	TGL 8728
R 218	1,5 kOhm	5 %	250.207 TK	TGL 8728
R 221	15 kOhm	5 %	250.207 TK	TGL 8728
R 222	10	5 %	250.207 TK	TGL 8728
R 223	47 kOhm	5 %	250.207 TK	TGL 8728
R 225, R 228	6,8 kOhm	5 %	250.207 TK	TGL 8728
R 226, R 227, R 235	470	5 %	250.207 TK	TGL 8728
R 229	100 kOhm	5 %	250.207 TK	TGL 8728

Tafel A 2.17.3.: Stückliste RAM/ROM-Karte

Schaltkreise

IS 301...IS 303, IS 305...IS 311, IS 328, IS 329	D 100		TGL 26152
IS 304	D 204		TGL 29263
IS 312...IS 319	U 501 D		TGL 32057
IS 320...IS 327	CM 8001		VRB
IS 330, IS 331	D 103		TGL 27148

Dioden

D 301...D 308	SAY 32		TGL 200-8466

Kondensatoren

C 301, C 302	100/6	Tantal-Kondensatoren 100/6	TGL 200-8519

Widerstände

R 301...R 321	1 kOhm	5 %	250.207 TK	TGL 8728
R 322	3 kOhm	5 %	250.207 TK	TGL 8728

Tafel A 2.17.4.: Stückliste IN/OUT-Karte

Schaltkreise

IS 401...IS 408	D 153	TGL 27148
IS 409, IS 410	D 103	TGL 27148
IS 411, IS 412, IS 415+, IS 416++, IS 418+...IS 420+, IS 429... IS 436	D 100	TGL 26152

+ nur bei IN/OUT-Karte 1
++ nur bei IN/OUT-Karte 2

Kondensatoren

C 401...C 415	47 nF	Kondensator SDVU 47 nF/50/63	7619.84
C 416	100/6	Tantal-Kondensator 100/6	TGL 200-8519

Widerstände

R 401, R 402	1 kOhm	5 %	250.207 TK	TGL 8728

Brücken

Br 401

Br 402

Tafel A 2.17.5.: Stückliste OUT-Karte

Schaltkreise

IS 501, IS 503, IS 505, IS 506, IS 508, IS 510, IS 512, IS 514, IS 516, IS 518, IS 520, IS 522, IS 524, IS 526, IS 528, IS 530...IS 533, IS 535, IS 537, IS 540, IS 541	D 100		TGL 26 152
IS 502, IS 504, IS 507, IS 509, IS 511, IS 513, IS 515, IS 517, IS 519, IS 521, IS 523, IS 525, IS 527, IS 529, IS 534, IS 536	D 195		TGL 28 467
IS 538, IS 539	D 110		TGL 26 152

Kondensatoren

C 501...C 519	47 nF	Kondensator SDVU 47 nF/50/63	76 19.84
C 520	100/6	Tantal-Kondensator 100/6	TGL 200-8519

Widerstände

R 501...R 503	1 kOhm	5 % 250.207 TK	TGL 8728

Tafel A 2.17.6.: Stückliste INT-Karte

Schaltkreise			
IS 601... IS 610, IS 624... IS 626	D 100		TGL 26 152
IS 611... IS 618, IS 620, IS 621	D 174		TGL 29 266
IS 619	D 130		TGL 26 152
IS 622, IS 623	D 120		TGL 26 152

Kondensatoren			
C 601... C 603	47 nF	Kondensator SDVU 47 nF/50/63	76 19.84
C 604	22 µ	Tantal-Kondensator	TGL 200-8519

Widerstände			
R 600... R 634	1 kOhm 5 %	250.207 TK	TGL 8728

Tafel A 2.17.7.: Stückliste A/D-Karte

Schaltkreise

IS 701...IS 708, IS 717, IS 723	A 109 D		TGL 28 873
IS 709...IS 716	A 110 D		TGL 28 873
IS 718, IS 719	D 103 D		TGL 27 148
IS 720...IS 722	D 100 D		TGL 26 152
IS 724, IS 725	36-28213		KWH

Transistoren

T 701	SC 237 C		TGL 11 811
T 702	SF 126 E		TGL 11 811
T 703...T 722	SF 137 D		TGL 11 811

Dioden

D 701	SZY 23		TLAB
D 702	SZX 19/5,1		TGL 200-8142
D 703, D 706	SZ 600/6,2		
D 704, D 705 D 707, D 708	SA 412		TGL 28 364

Kondensatoren

C 701, C 704, C 707 C 710, C 713, C 716 C 719, C 722	2,2 µF/20 %	Lackkondensator MKL 3	TGL 10 793/03
C 702, C 705, C 708, C 711, C 714, C 717, C 720, C 723, C 725, C 728	220 pF/10 %	Scheibenkondensator EDVU N 1500	TGL 24 100
C 703, C 706, C 709, C 712, C 715, C 718, C 721, C 724, C 726, C 727	4,7 nF/10 %	Scheibenkondensator EDVU E 2000	TGL 24 100
C 729, C 732, C 737	100 µF/10 V	Elektrolytkondensator Typ IIA	TGL 200-8308
C 730, C 731, C 733, C 734, C 735, C 736, C 738, C 739	47 µF/16 V	Elektrolytkondensator Typ IIA	TGL 200-8308
C 740...C 773	15 nF/20 %	Scheibenkondensator SDVU	3312.4-7300.8:

Widerstände

R 701, R 703, R 705, R 707, R 709, R 711, R 713, R 715,	65 kOhm	1 %	250.311	TK 50	TGL 8728
R 702, R 704, R 706, R 708, R 710, R 712, R 714, R 716,	18 kOhm	10 %	250.311	TK 50	TGL 8728
R 717, R 718, R 787, R 788,	1,2 kOhm	5 %	250.311	TK 100	TGL 8728
R 719, R 720, R 785, R 786,	43 Ohm	5 %	250.311	TK 100	TGL 8728
R 721, R 724, R 727, R 730, R 733, R 736, R 739, R 742,	27 kOhm	1 %	250.311	TK 50	TGL 8728
R 722, R 725, R 728, R 731, R 734, R 737, R 740, R 743, R 745,	51 Ohm	10 %	250.311	TK 200	TGL 8728
R 723, R 726, R 729, R 732, R 735, R 738, R 741, R 744, R 747, R 762, R 763, R 765, R 767, R 769, R 771, R 773, R 775, R 777, R 779, R 781,	1,5 kOhm	10 %	250.311	TK 200	TGL 8728
R 746	1,5 kOhm	0,25 %	250.311	TK 50	TGL 8728
R 748	710 Ohm	10 %	250.311	TK 50	TGL 8728
R 749...R 756	8,2 kOhm	10 %	250.311	TK 200	TGL 8728
R 757	560 Ohm	10 %	250.311	TK 50	TGL 8728
R 758	3 kOhm	1 %	250.311	TK 50	TGL 8728
R 759	5,1 kOhm	10 %	250.311	TK 50	TGL 8728
R 760	1,2 kOhm	10 %	250.311	TK 50	TGL 8728
R 761	1 kOhm	5 %	250.311	TK 50	TGL 8728
R 764, R 766, R 768, R 770, R 772, R 774, R 776, R 778, R 780, R 782	3,6 kOhm	10 %	250.311	TK 200	TGL 8728
R 783, R 784,	160 Ohm	10 %	250.412	TK 200	TGL 8728

Einstellregler

P 701, P 703, P 705, P 707, P 709, P 711, P 713, P 715,	4,7 kOhm	Dick-schicht SN 3	0120.601-00001	TGL 27 423

Einstellregler

P 702, P 704, P 706, P 708, P 710, P 712, P 714, P 716, P 717,	100 Ohm	Dickschicht SN 3	0120.601-00001	TGL 27 423
P 718	1 kOhm	Dickschicht SN 3	0120.601-00001	TGL 27 423

Tafel A 2.17.8.: Stückliste NT-Karte

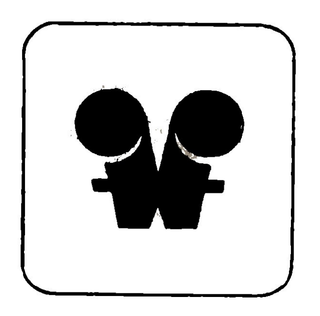
Schaltkreise			
IS 801...IS 804		MAA 723	
Transistoren			
T 801, T 804, T 808, T 809,		KF 517 b	
T 802, T 810,		KU 602	
T 803		2 N 3055	
T 805		KU 601	
T 806		SF 128	TGL 200-8439
T 807		KU 607	
Dioden			
D 801...D 804, D 807...D 810, D 816...D 819,		SY 320/2	TGL 28 818
Thyristoren			
Th 801, Th 802		ST 103/1	TGL 28 220
Kondensatoren			
C 801, C 813	470/40	Elektrolytkondensator	TGL 7198
C 802	100/35	Elektrolytkondensator	TGL 26 629
C 803, C 806, C 810, C 814	10 nF	Kondensator SDVU 10/50/63	71 19.84
C 804, C 808, C 809	4,7 nF	EDVU-Kondensator E 2000	TGL 24 100
C 805	100 nF	Kondensator SDVU 100/50/63	78 19.84
C 807, C 811, C 815	15 nF	Kondensator SDVU 15/50/63	73 19.84
C 812	47 nF	Kondensator SDVU 47/50/63	76 19.84
C 817...C 820	5000/25	Elektrolytkondensator	TGL 5151
C 821	0,1+2x2500	Funkentstörkondensator	TGL 11 840

Widerstände

R 801	10 Ohm	5 %	250.207 TK	TGL 8728
R 802, R 806	1 kOhm	5 %	250.207 TK	TGL 8728
R 803	2 kOhm	Einstellregler P		TGL 11 886
R 804	1,8 kOhm	5 %	250.207 TK	TGL 8728
R 805, R 807, R 808, R 812, R 816, R 818,	68 Ohm	5 %	250.207 TK	TGL 8728
R 809, R 811, R 815, R 822,	6,8 kOhm	5 %	250.207 TK	TGL 8728
R 810, R 814, R 821,	2,5 kOhm	5 %	250.207 TK	TGL 8728
R 813	220 Ohm	5 %	250.207 TK	TGL 8728
R 817	0,3 Ohm			
R 819, R 820	3,3 kOhm	5 %	250.207 TK	TGL 8728
R 823	1,2 kOhm	5 %	250.207 TK	TGL 8728
R 824	2,4 kOhm	5 %	250.207 TK	TGL 8728

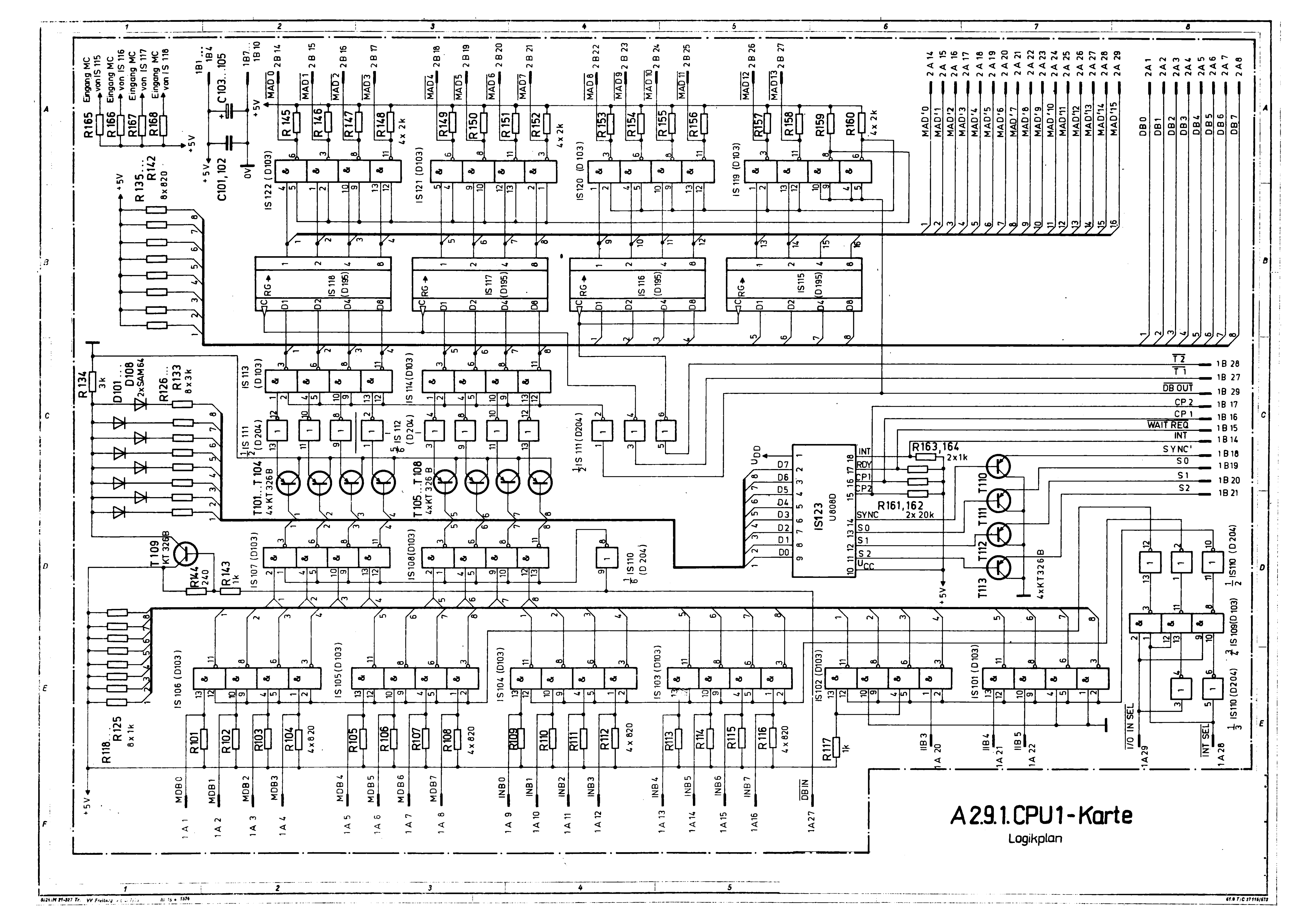
Sicherung

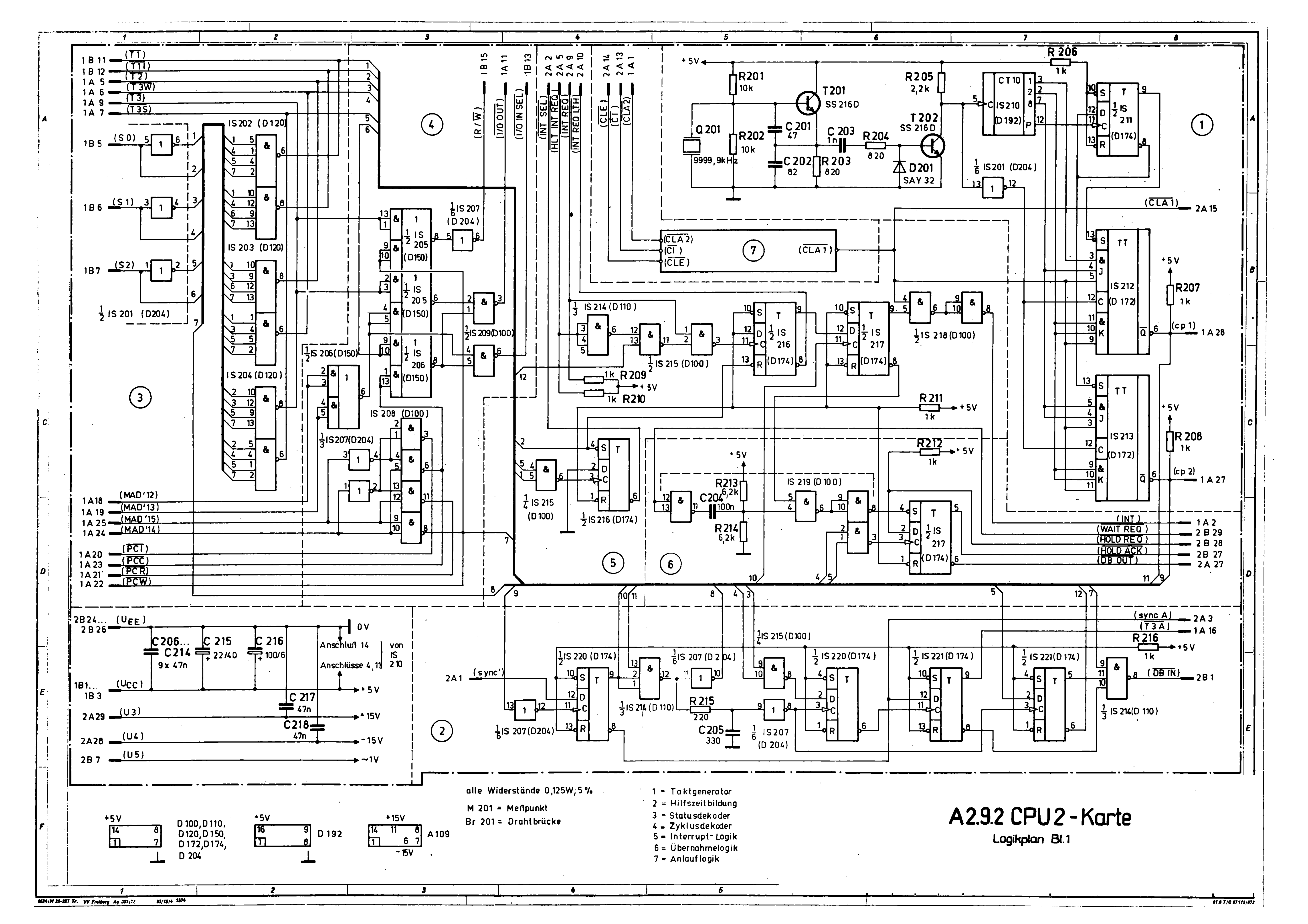
Si 801	T 6,3 A	G-Schmelzeinsatz	TGL 0-41 571



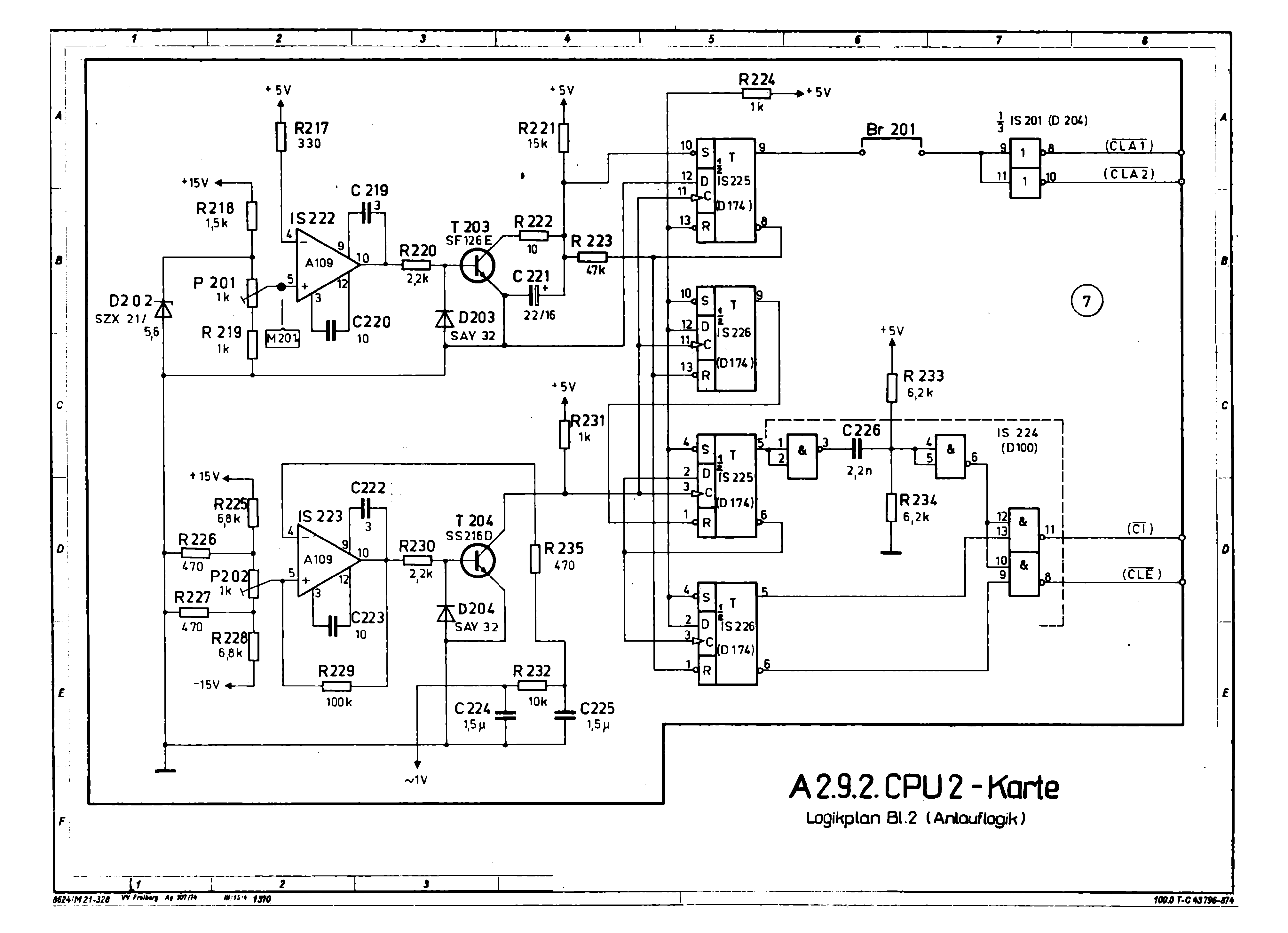
HEIM-ELECTRIC
EXPORT-IMPORT
VOLKSEIGENER AUSSENHANDELSBETRIEB DER
DEUTSCHEN DEMOKRATISCHEN REPUBLIK
DDR-1026 BERLIN·ALEXANDERPLATZ
HAUS DER ELEKTROINDUSTRIE

KOMBINAT
VEB FUNKWERK ERFURT
DDR - 501 Erfurt, Rudolfstraße 47
Telefon: 5 80
Telex: 061 306
Kabel: FUNKWERK ERFURT


A 2.9.1. CPU1-Karte
Logikplan
IS 123
U 808 D
IS 115 (D195)
IS 116 (D195)
IS 117 (D195)
IS 118 (D195)
IS 119 (D103)
IS 120 (D103)
IS 121 (D103)
IS 122 (D103)
IS 113 (D103)
IS 114 (D103)
IS 111 (D204)
IS 112 (D204)
IS 107 (D103)
IS 108 (D103)
IS 110 (D204)
IS 109 (D103)
IS 101 (D103)
IS 102 (D103)
IS 103 (D103)
IS 104 (D103)
IS 105 (D103)
IS 106 (D103)
T101...T104 4x KT 326 B
T105...T108 4x KT 326 B
T 109 KT 326B
4xKT 326 B
D101...D108 2xSAM64
R 165 Eingang MC von IS 115
R 166 Eingang MC von IS 116
R 167 Eingang MC von IS 117
R 168 Eingang MC von IS 118
C 103...105
C101, 102
R 135...R142 8x820
R126...R133 8x3k
R 134 3k
R118...R125 8x1k
R161,162 2x 20k
R163,164 2x1k
R 117 1k
T2
T1
DB OUT
CP 2
CP 1
WAIT REQ
INT
SYNC'
S0
S1
S2
1B 28
1B 27
1B 29
1B 17
1B 16
1B 15
1B 14
1B 18
1B 19
1B 20
1B 21
MDB0
MDB1
MDB2
MDB3
MDB4
MDB5
MDB6
MDB7
INB0
INB1
INB2
INB3
INB4
INB5
INB6
INB7
DBIN
I/O IN SEL
INT SEL
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7



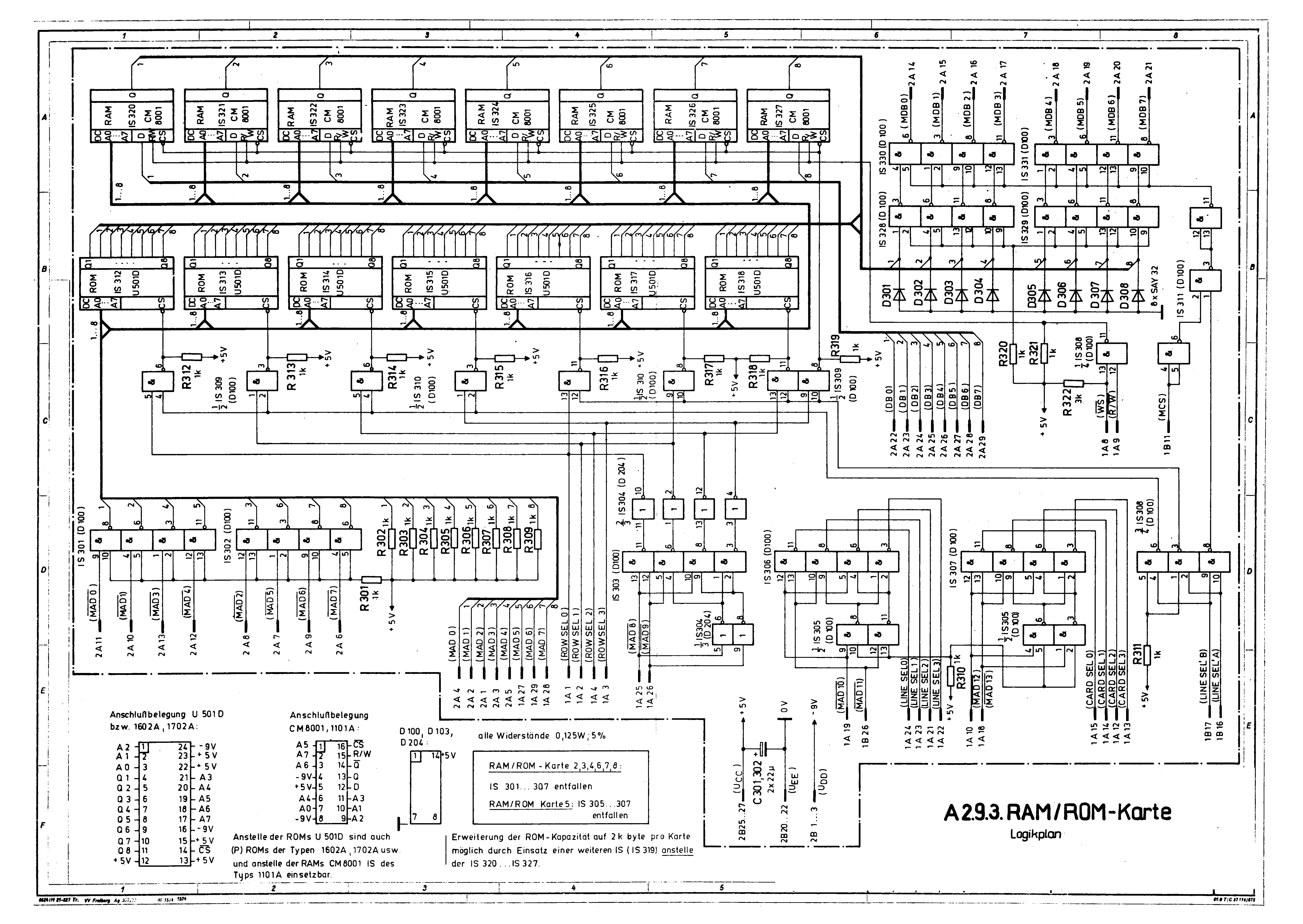
alle Widerstände 0,125W; 5%
M 201 = Meßpunkt
Br 201 = Drahtbrücke
1 = Taktgenerator
2 = Hilfszeitbildung
3 = Statusdekoder
4 = Zyklusdekoder
5 = Interrupt-Logik
6 = Übernahmelogik
7 = Anlauflogik
+5V 14 8 1 7 D 100, D110, D120, D150, D172, D174, D 204
+5V 16 9 1 8 D 192
+15V 14 11 8 1 6 7 -15V A 109
Anschluß 14 von IS 210
Anschlüsse 4, 11
A2.9.2 CPU 2 - Karte
Logikplan Bl. 1



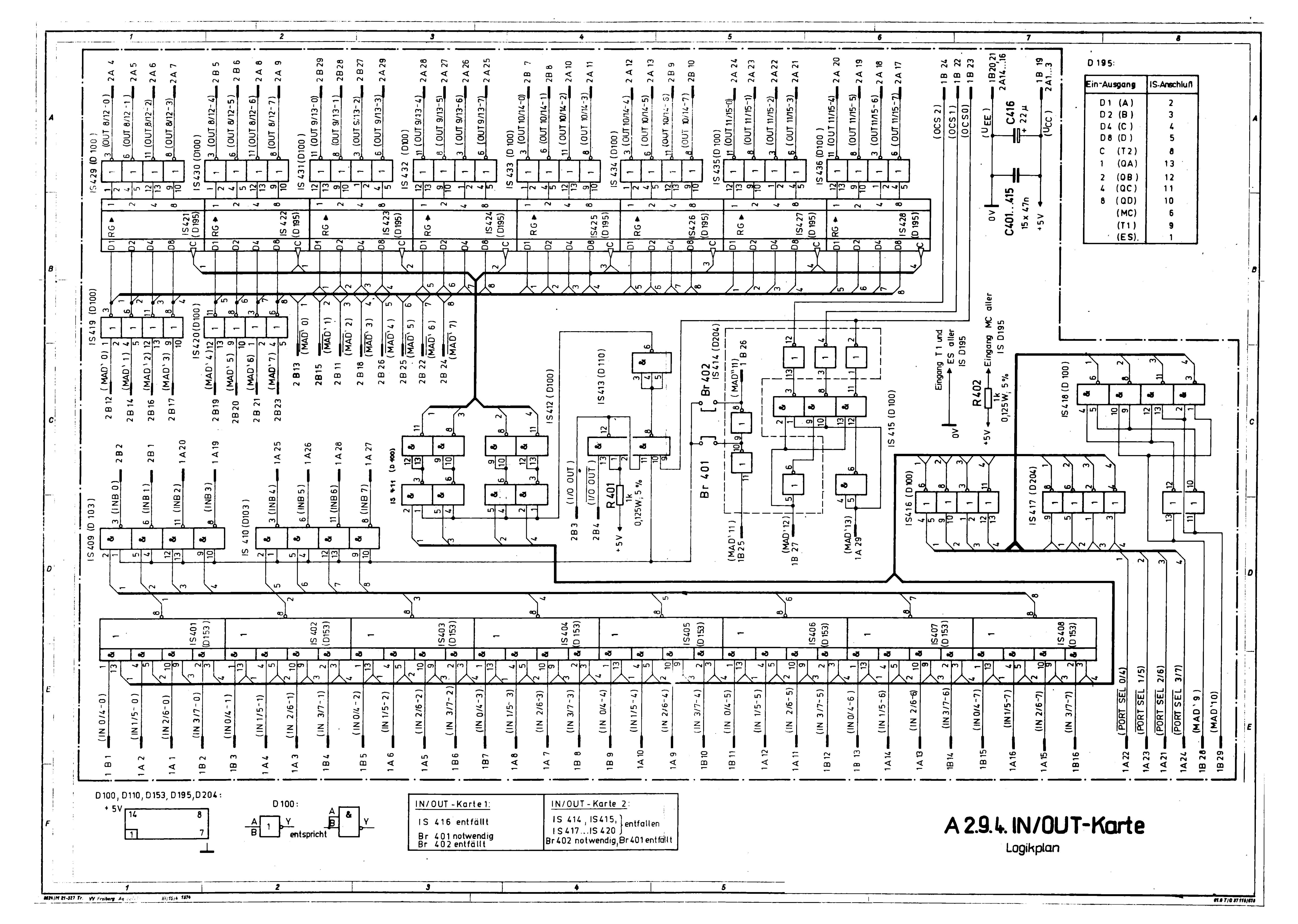
A 2.9.2. CPU 2 - Karte

Logikplan Bl.2 (Anlauflogik)

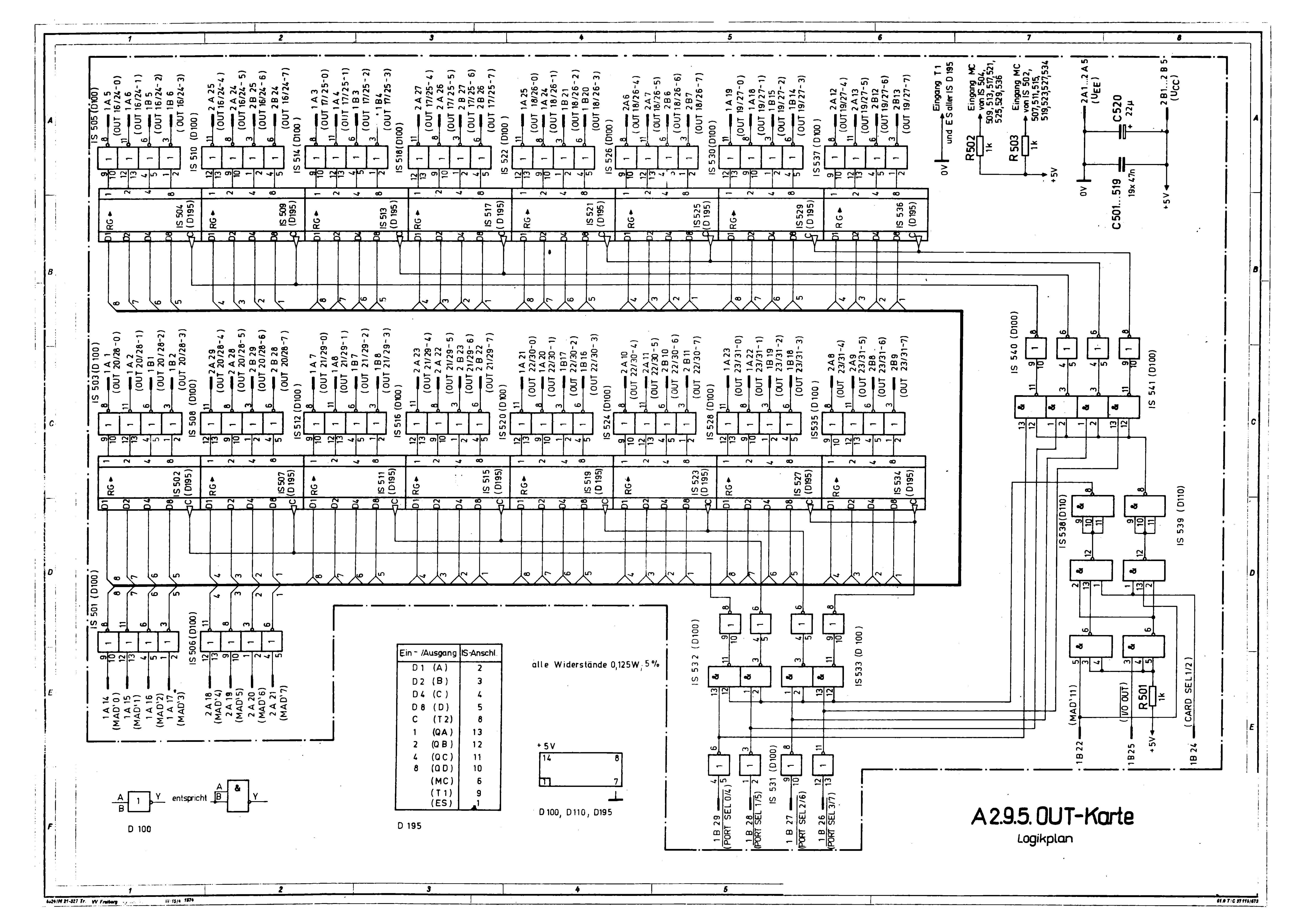
8624/M 21-328 VV Freiberg Ag 307/74 III:15:4 1370 100.0 T-C 43796-874



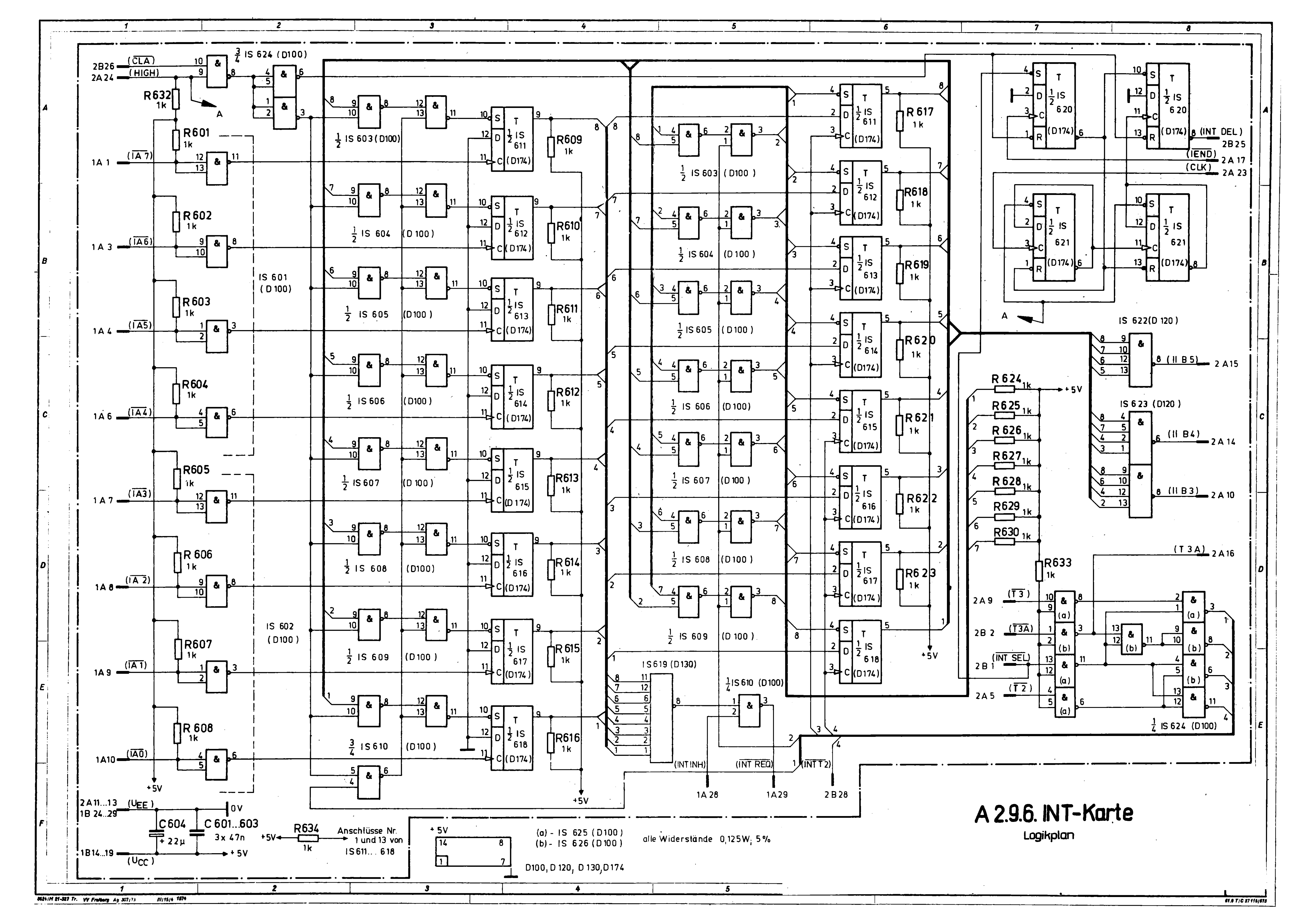
A2.9.3. RAM/ROM-Karte
Logikplan
Anschlußbelegung U 501D bzw. 1602A, 1702A:
A2 1 24 -9V
A1 2 23 +5V
A0 3 22 +5V
Q1 4 21 A3
Q2 5 20 A4
Q3 6 19 A5
Q4 7 18 A6
Q5 8 17 A7
Q6 9 16 -9V
Q7 10 15 +5V
Q8 11 14 CS
+5V 12 13 +5V
Anschlußbelegung CM8001, 1101A:
A5 1 16 CS
A7 2 15 R/W
A6 3 14 Q
-9V 4 13 Q
+5V 5 12 D
A4 6 11 A3
A0 7 10 A1
-9V 8 9 A2
D 100, D 103, D 204:
alle Widerstände 0,125W; 5%
RAM/ROM-Karte 2,3,4,6,7,8: IS 301...307 entfallen
RAM/ROM Karte 5: IS 305...307 entfallen
Anstelle der ROMs U 501D sind auch (P) ROMs der Typen 1602A, 1702A usw. und anstelle der RAMs CM 8001 IS des Typs 1101A einsetzbar.
Erweiterung der ROM-Kapazität auf 2k byte pro Karte möglich durch Einsatz einer weiteren IS (IS 319) anstelle der IS 320...IS 327.
C301,302 2x22µ
8x SAY 32



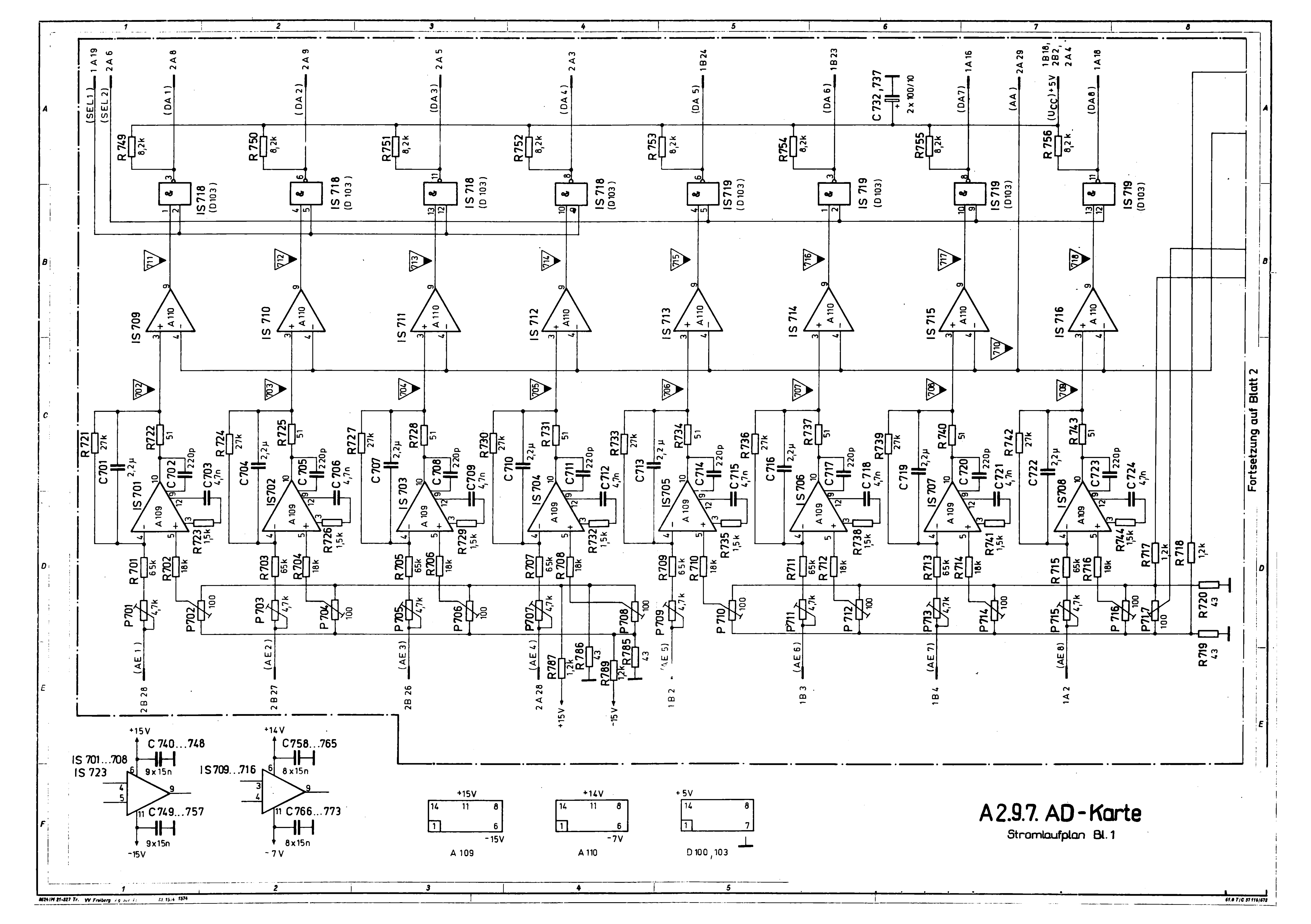
A 2.9.4. IN/OUT-Karte
Logikplan
D 195:
Ein-Ausgang | IS-Anschluß
D1 (A) 2
D2 (B) 3
D4 (C) 4
D8 (D) 5
C (T2) 8
1 (QA) 13
2 (QB) 12
4 (QC) 11
8 (QD) 10
(MC) 6
(T1) 9
(ES) 1
D100, D110, D153, D195, D204:
+5V 14 8
1 7
D100:
entspricht
IN/OUT-Karte 1:
IS 416 entfällt
Br 401 notwendig
Br 402 entfällt
IN/OUT-Karte 2:
IS 414, IS415, IS417...IS 420 entfallen
Br 402 notwendig, Br 401 entfällt
C416 + 22 µ
C401...415 15 x 47n
R 401 1k 0,125W, 5 %
R 402 1k 0,125W, 5 %
Eingang T1 und ES aller IS D195
Eingang MC aller IS D195
Br 401
Br 402



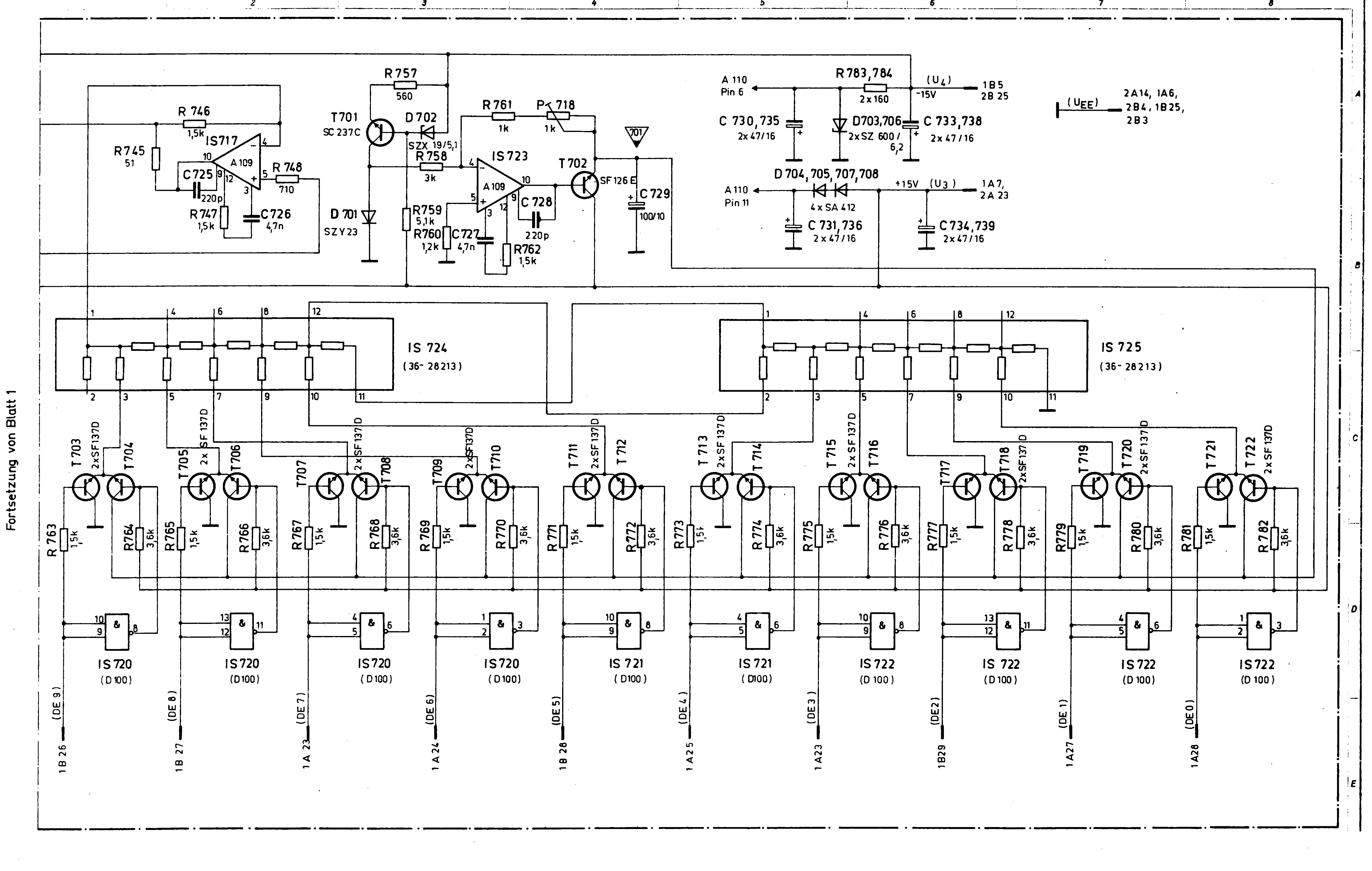
A 2.9.5. OUT-Karte
Logikplan
alle Widerstände 0,125W; 5 %
Ein- /Ausgang | IS-Anschl.
D 1 (A) | 2
D 2 (B) | 3
D 4 (C) | 4
D 8 (D) | 5
C (T2) | 8
1 (QA) | 13
2 (QB) | 12
4 (QC) | 11
8 (QD) | 10
(MC) | 6
(T1) | 9
(ES) | 1
D 195
D 100, D110, D195
+5V
entspricht
D 100
Eingang T1 und ES aller IS D 195
Eingang MC von IS 504, 509, 513, 517, 521, 525, 529, 536
Eingang MC von IS 502, 507, 511, 515, 519, 523, 527, 534
R502 1k
R 503 1k
C520 22µ
C501...519 19x 47n
2A1...2A5 (UEE)
2B1...2B5 (UCC)
R 501 1k
1B 22 (MAD'11)
1B25 (I/O OUT)
1B 24 (CARD SEL 1/2)
1B 29 (PORT SEL 0/4)
1B 28 (PORT SEL 1/5)
1B 27 (PORT SEL 2/6)
1B 26 (PORT SEL 3/7)
IS 531 (D100)
IS 532 (D100)
IS 533 (D 100)
IS 538 (D110)
IS 539 (D110)
IS 540 (D100)
IS 541 (D100)
IS 501 (D100)
IS 506 (D100)
1 A 14 (MAD'0)
1 A 15 (MAD'1)
1 A 16 (MAD'2)
1 A 17 (MAD'3)
2 A 18 (MAD'4)
2 A 19 (MAD'5)
2 A 20 (MAD'6)
2 A 21 (MAD'7)



A 2.9.6. INT-Karte
Logikplan
3/4 IS 624 (D100)
2B26 (CLA)
2A24 (HIGH)
R632 1k
IS 601 (D100)
IS 602 (D100)
1/2 IS 603 (D100)
1/2 IS 604 (D100)
1/2 IS 605 (D100)
1/2 IS 606 (D100)
1/2 IS 607 (D100)
1/2 IS 608 (D100)
1/2 IS 609 (D100)
3/4 IS 610 (D100)
1/2 IS 611 (D174)
1/2 IS 612 (D174)
1/2 IS 613 (D174)
1/2 IS 614 (D174)
1/2 IS 615 (D174)
1/2 IS 616 (D174)
1/2 IS 617 (D174)
1/2 IS 618 (D174)
1/2 IS 620 (D174)
1/2 IS 621 (D174)
IS 622 (D120)
IS 623 (D120)
IS619 (D130)
1/4 IS610 (D100)
1/4 IS 624 (D100)
(INT DEL) 2B25
(IEND) 2A17
(CLK) 2A23
(II B5) 2A15
(II B4) 2A14
(II B3) 2A10
(T3A) 2A16
2A9 (T3)
2B2 (T3A)
2B1 (INT SEL)
2A5 (T2)
(INT INH) 1A28
(INT REQ) 1A29
(INT T2) 2B28
1A1 (IA7)
1A3 (IA6)
1A4 (IA5)
1A6 (IA4)
1A7 (IA3)
1A8 (IA2)
1A9 (IA1)
1A10 (IA0)
2A11...13 (UEE) 1B24...29
0V
C604 +22µ
C601...603 3x47n
1B14...19 (UCC) +5V
R634 1k
Anschlüsse Nr. 1 und 13 von IS611 ... 618
(a) - IS 625 (D100)
(b) - IS 626 (D100)
alle Widerstände 0,125W; 5%
D100, D120, D130, D174

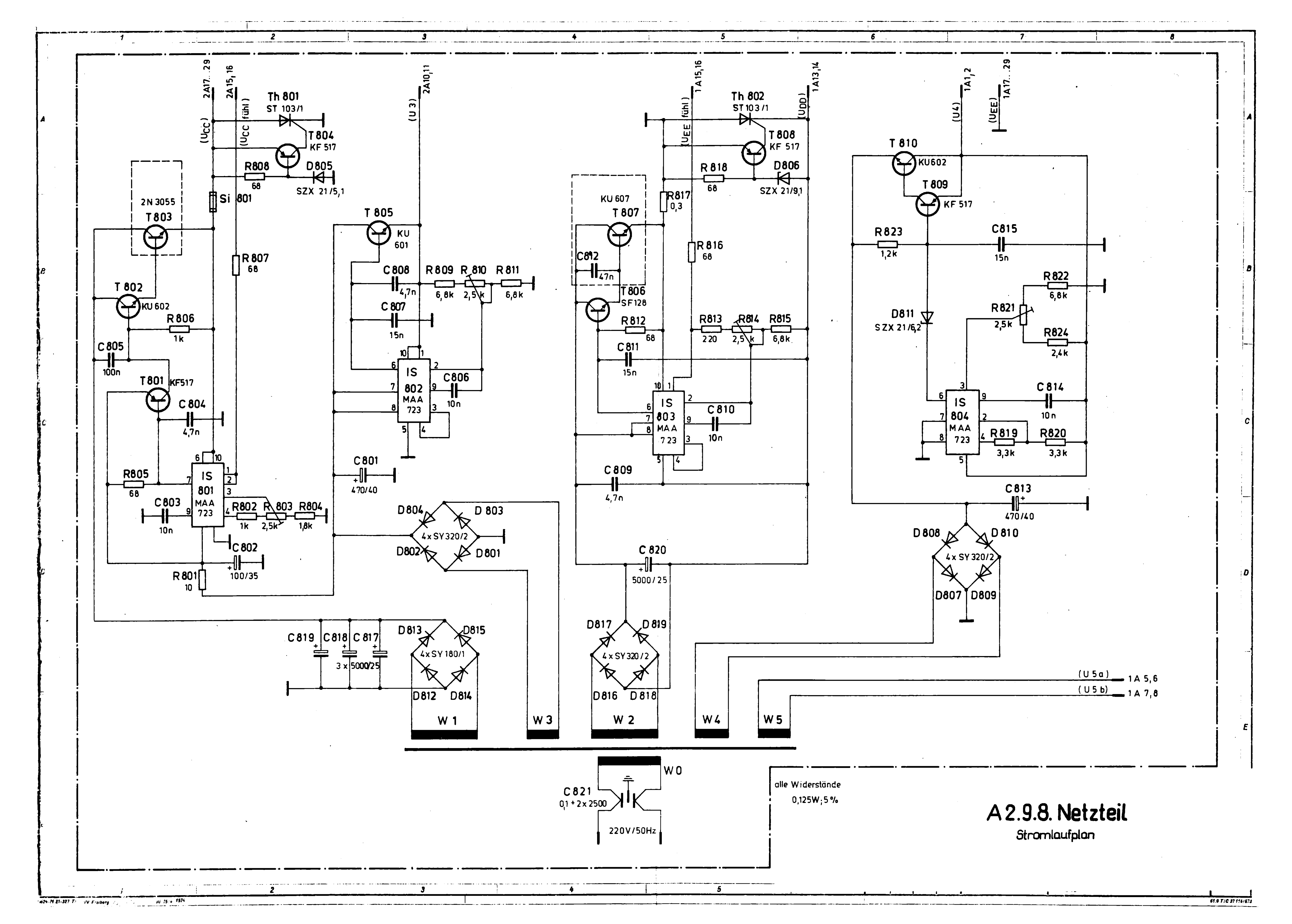


A 2.9.7. AD - Karte
Stromlaufplan Bl. 1
Fortsetzung auf Blatt 2
IS 701...708
IS 723
IS 709...716
C 740...748
C 749...757
C 758...765
C 766...773
A 109
A 110
D 100, 103



A 2.9.7. AD-Karte

Stromlaufplan Bl. 2



2A17...29
2A15,16
(UCC)
(UCC fühl)
Th 801
ST 103/1
T 804
KF 517
R808
68
D805
SZX 21/5,1
Si 801
2N 3055
T 803
R 807
68
T 802
KU 602
R 806
1k
C 805
100n
T 801
KF517
C 804
4,7n
R805
68
IS
801
MAA
723
C 803
10n
R802
1k
R 803
2,5k
R804
1,8k
C 802
100/35
R 801
10
2A10,11
(U 3)
T 805
KU
601
C 808
4,7n
R 809
6,8k
R 810
2,5 k
R 811
6,8k
C 807
15n
IS
802
MAA
723
C 806
10n
C 801
470/40
D 804
D 803
4 x SY 320/2
D802
D 801
1A 15,16
(UEE fühl)
Th 802
ST 103/1
(UDD)
1A 13,14
T 808
KF 517
R 818
68
D806
SZX 21/9,1
KU 607
T 807
R817
0,3
R 816
68
C 812
47n
T 806
SF 128
R 812
68
R813
220
R 814
2,5 k
R 815
6,8 k
C 811
15n
IS
803
MAA
723
C 810
10n
C 809
4,7n
C 820
5000/25
1A 1,2
(U4)
1A 17...29
(UEE)
T 810
KU602
T 809
KF 517
R 823
1,2k
C 815
15n
R 822
6,8k
D 811
SZX 21/6,2
R 821
2,5k
R 824
2,4k
IS
804
MAA
723
C 814
10 n
R 819
3,3k
R 820
3,3k
C 813
470/40
D 808
D 810
4 x SY 320/2
D807
D809
C 819
C 818
C 817
3 x 5000/25
D 813
D815
4 x SY 180/1
D 812
D 814
D 817
D 819
4 x SY 320/2
D 816
D 818
(U 5 a)
1A 5,6
(U 5 b)
1A 7,8
W 1
W 3
W 2
W 4
W 5
W 0
C 821
0,1 + 2 x 2500
220V/50Hz
alle Widerstände
0,125W; 5 %
A 2.9.8. Netzteil
Stromlaufplan